

BACnet Driver

Filename	BACnet.dll
Manufacturer	ASHRAE
Devices	Any device implementing BACnet/IP and BACnet MS/TP protocols
Protocol	BACnet/IP (Master) and BACnet MS/TP
Version	1.2.117
Last Update	12/12/2024
Platform	Win32
Dependencies	IOKit 2.00
Superblock Reading	No
Level	0

Introduction

This Driver implements the BACnet/IP (Master) and BACnet MS/TP communication protocols by ASHRAE, which allows **Eclipse Software** products to communicate with any device implementing BACnet/IP (Master) and BACnet MS/TP protocols.

The BACnet Driver was developed using **Eclipse Software's IOKit** library. For more information about this library, please check topic **Documentation of I/O Interfaces**.

Driver Configuration

This Driver's [P] parameters are not used and must be left with the value 0 (zero). Specific settings of this Driver must be configured using the **BACnet** tab on the configuration window, available on **Driver settings** in **Eclipse E3** or **Eclipse Power**. In **Eclipse SCADA**, select the Driver and click **Extra** in Organizer.

The **Setup**, **Serial**, **Ethernet**, **Modem**, and **RAS** tabs contain the configuration of **IOKit** library. For more information, please check topic **Documentation of I/O Interfaces**.

BACnet Tab

Driver BACnet v1.2.112 (IOKit v2.0.142)

BACnet
Setup
Serial
Ethernet
Modem
RAS

Protocol: BACnet IP (UDP)

☐ Discover device addresses dynamically
☒ Use the following device list:

Property	Value
General	
Log Detailed BACnet Packets	False
Auto-Detect Devices Range	0-4194302
Application Timeout (milliseconds)	2000
Wait Timeout (milliseconds)	10000
Retries	2
Disable Read Grouping	False
Maximum Transactions per Network	1
Network Delay (milliseconds)	0
COV Process Identifier	0
Read Quality-related Properties	True
Driver Device Instance	191
Driver Device Name	Elipse BACnet Driver #191
Device Defaults	
Maximum number of properties per...	50
Inter-Message Delay (milliseconds)	0
Use COV	True
Use Confirmed COV	False
Maximum COV Subscriptions	0
Exclude Analog objects from COV	False
COV Subscription Lifetime (seconds)	0

Device	IP	Subnet/MAC	Use COV	Read Multiple	Write Priority	Alarms

Add
Edit
Delete
Auto-detect...
Use All IPs
Use this IP

Current IP: (all)

OK
Cancel
Apply

BACnet tab

The available options on this tab are described on the next table.

Available options on the BACnet tab

OPTION	DESCRIPTION
Protocol	Defines the protocol used by this Driver. The available items are BACnet MS/TP (RS485) and BACnet IP (UDP) , which is automatically selected. Future versions of this Driver must also add the BACnet PTP protocol
Properties	Defines specific properties of this Driver. Those properties are separated into groups General and Device Defaults , described next
Discover device addresses dynamically	Select this option so that this Driver does not use a fixed list of devices. Therefore, devices can be added dynamically by an application at run time
Use the following device list	Select this option so that this Driver uses a fixed list of devices. With this option selected, user can either click Auto-detect or fill in the list of devices manually. At run time, only the devices declared on this list are used. The columns on this list are described on the table Columns on the list of devices
Add	Adds a new device on the list

OPTION	DESCRIPTION
Edit	Allows changing the configuration of the selected device on the list. Please check topic Add or Edit a BACnet device for more information about this option
Delete	Removes the selected device from the list
Auto-detect	Searches for BACnet devices available on the local network. Clicking this option removes the content of the list and fills it with all devices found. Please check topic Auto-Detecting Devices for more information
Use All IPs	Indicates that this Driver must send Tags to all IP addresses from the list
Use this IP	Indicates that this Driver must send Tags only to the selected IP address. This allows several instances of this Driver to send different Tags to different IP addresses

The **Properties** list contains the options described on the next tables.

Available options on the General group

OPTION	DESCRIPTION
Log Detailed BACnet Packets	Enable this option so that this Driver generates, on the log file, detailed information about the received and sent BACnet packets
Auto-Detect Devices Range	Range of searched devices through auto-detection services. Please check topic Auto-Detecting Devices for more information. The default value for this property is "0-4194302"
Application Timeout (milliseconds)	Time-out for transactions with devices, in milliseconds
Wait Timeout (milliseconds)	Time-out for the link layer, in milliseconds. Default value of this option is 10000, or 10 seconds. This is the time this Driver waits to clear the communication channel. For example, when performing a writing to a device, this writing process may wait for an ongoing reading process to finish. This option defines a maximum time for this time-out. If the channel is not available after waiting for this configured time, this Driver fails this operation
Retries	Allows defining a number of retries for requests that failed due to communication errors
Disable Read Grouping	Select this option so that this Driver does not use grouped readings, or readings of several Tags on the same request, nor notifications of value changes
Maximum Transactions per Network	Maximum number of simultaneous transactions on the same network. The network number of a device is determined by the Enable Network Address , Network Address , and MAC Address properties. The default value of this property is 1 (one) and the maximum value is 100
Network Delay (milliseconds)	Delay, in milliseconds, between messages sent to the same external network. This delay is not applied to messages to

OPTION	DESCRIPTION
	local or broadcast devices. The network number of a device is determined by the Enable Network Address , Network Address , and MAC Address properties. The default value of this property is 0 (zero)
COV Process Identifier	Identifier of the process executing the COV System
Read Quality-related Properties	When this option is enabled, Tags reading the PresentValue property of an object also implicitly read that object's Reliability and StatusFlags properties, which are used to determine the quality of a Tag. When this option is disabled, the quality of a Tag is determined by these properties only if they are explicitly read by other Tags. The default value for this property is True (enabled)
Driver Device Instance	Driver Device identifier. This number shall be unique among all BACnet internetwork-wide devices. This value is associated to the objectIdentifier property of the Device object. The default value for this property is 191 and the maximum value is 4194302
Driver Device Name	Driver name. This name shall be unique among all BACnet internetwork-wide devices. This value is associated to the objectName property of the Device object. The default value for this property is "Eclipse BACnet Driver #191"

Available options on the Device Defaults group

OPTION	DESCRIPTION
Maximum number of properties per ReadPropertyMultiple request	Maximum number of properties present in a multiple reading request sent by this Driver
Inter-Message Delay (milliseconds)	Delay, in milliseconds, between messages sent to the same device. Default value of this option is 0 (zero)
Use COV	Enables the system of spontaneous messages (<i>Change of Value</i> , or COV), sent by a device when a value changes. Please check topic COV System for more information
Use Confirmed COV	Configures a remote device to request a receive confirmation for COV (<i>Change Of Value</i>) packets. Used only when the Use COV option is equal to True and the Disable Read Grouping option is equal to False
Maximum COV Subscriptions	Maximum number of COV (<i>Change Of Value</i>) subscriptions this Driver establishes per device. When reaching this limit, all other Tags are read by polling
Exclude Analog objects from COV	When this option is enabled, readings of analog values are not performed by COV notifications
COV Subscription Lifetime (seconds)	Time, in seconds, in which a COV subscription is valid. A value of 0 (zero) indicates that a subscription does not expire. This Driver renews expired subscriptions automatically
SubscribeCOV Delay (milliseconds)	Delay, in milliseconds, between sending a request for a COV subscription and the next message to the same device. Default value of this option is 0 (zero)

The columns on the list of devices are described on the next table.

Columns on the list of devices

COLUMN	DESCRIPTION
Device	The identifier of this BACnet device. The icon ✓ indicates this device can be accessed by the currently configured IP address and the icon ✗ indicates this device is on an IP address different from the one configured in IOKit
IP	Shows the IP address where this device is. An instance of BACnet Driver can access only one IP address. This list's footer displays the currently configured IP address
Subnet/MAC	Shows the network and MAC address for this device, if enabled
Use COV	Shows COV (<i>Change Of Value</i>) settings for this device
Read Multiple	Shows the maximum number of properties present in a multiple reading request sent to this device
Write Priority	Shows the value of the priority to be used in writing requests for this device, if enabled
Alarms	Shows alarm settings for this device

Add or Edit a BACnet Device

When clicking **Add** or **Edit** on the **BACnet** tab the window on the next figure is opened.

Add/Edit BACnet device

Property	Value
Device Instance	0
Segmentation Support	Both
Segments Accepted	Unspecified
Maximum APDU Length Accepted	Up to 1476 octets (fits in an Ethernet frame)
Proposed Window Size for Segmentation	10
Maximum number of properties per ReadProp...	50
Inter-Message Delay (milliseconds)	0
Get Alarm Summary at startup	False
Network Address	
Device IP	
Enable Network Address	False
Network Address	0
MAC Address	
Write Priority	
Enable Write Priority	False
Write Priority	1
COV	
Use COV	True
Use Confirmed COV	False
Maximum COV Subscriptions	0
Exclude Analog objects from COV	False

OK
Cancel

Add/Edit BACnet device window

This window contains the properties described on the next table.

Available properties on the Add/Edit BACnet device window

PROPERTY	DESCRIPTION
Device Instance	Device's identifier
Segmentation Support	Type of packet segmentation supported by a device
Segments Accepted	Acceptable number of segments for exchanging segmented messages. Select accordingly to what is determined in a device's configuration
Maximum APDU Length Accepted	Maximum length number in octets per message segment. This means that if the request or response exceeds this maximum value, the message must be segmented into as many segments as necessary. Select accordingly to what is determined in a device's configuration
Maximum number of objects per ReadMultipleRequest	If the Disable Read Grouping option is deselected, this Driver performs grouped requests of several Tags that cannot be read spontaneously (COV). This option indicates the maximum number of items requested at the same time in these readings
Inter-Message Delay (milliseconds)	Delay, in milliseconds, between sent messages

PROPERTY	DESCRIPTION
Get Alarm Summary at startup	Performs a request to retrieve a list of alarms during this Driver's initialization. That retrieved list is then sent to the special Tag ReceiveEventNotifications
Device IP	IP address of a device
Enable Network Address	Select this option to allow editing a device's sub-network and MAC addresses
Network Address	Mask used on the IP address, which corresponds to the DNET field in a BACnet packet
MAC Address	MAC address of a device, which corresponds to the DADR field in a BACnet packet
Enable Write Priority	Select this option to allow specifying a writing priority to a device
Write Priority	Select this option to specify a priority used on all writing commands to properties sent to this device, between 1 (one) and 16. If this option is not selected, this Driver uses the default priority (16)
Use COV	Enables the system of spontaneous messages (<i>Change of Value</i> , or COV), sent by a device when the value changes. They are typically used for properties of type presentValue and StatusFlags . Please check topic COV System for more information
Use Confirmed COV	If enabled, informs that COV messages sent by a device must be confirmed. Otherwise, these messages do not need confirmation
Maximum COV Subscriptions	Informs a maximum number of variables that can be monitored by a device for spontaneous sending. After that limit, additional Tags are read by polling
Exclude Analog objects from COV	When this option is enabled, readings of analog values in this device are not performed by COV notifications
COV Subscription Lifetime (seconds)	Time, in seconds, in which a COV subscription for this device is valid. A value of 0 (zero) indicates that a subscription never expires. This Driver renews expired subscriptions automatically
SubscribeCOV Delay (milliseconds)	Delay, in milliseconds, between sending a request for a COV subscription and the next message

Auto-Detecting Devices

This Driver can search for BACnet devices in the local network. This is performed by broadcasting the **Who-Is** BACnet service and receiving the **I-Am** BACnet service. To do so, click **Auto-Detect** on the **BACnet** tab. If the list of devices is empty, this feature can also be started by expanding the **Network Browser** node in **Tag Browser**.

The **Auto-Detect Devices Range** property defines the search ranges to use by this Driver. This property must be filled with a list of ranges of instances of devices to search for. The syntax for each range is the following:

```
"<low device #>-<high device #>"
```

Or:

```
"<single device #>"
```

Where:

- **<low device #>**: Instance number of the beginning of the range, included, which must be less than **<high device #>**
- **<high device #>**: Instance number of the end of the range, included, which must be greater than **<low device #>**
- **<single device #>**: Instance number of the range with a single value

These ranges must be listed in ascending order, without intersections, and separated by commas. Examples:

- **"0-4194302"**: Searches for devices whose instance number ranges from 0 (zero) to 4194302. This is the property's default value, which corresponds to the entire range
- **"0-100, 200-300"**: Searches for devices whose instance number ranges from 0 (zero) to 100, or from 200 to 300
- **"0-100, 150, 200-300"**: Searches for devices whose instance number ranges from 0 (zero) to 100, is equal to 150, or from 200 to 300

Each defined range corresponds to sending a **Who-Is** message and receiving **I-Am** responses. Therefore, the more defined ranges, the longer the auto-detection process takes.

COV System

A **COV System** (*Change of Value*) allows devices to warn a Driver if there is a change to a specific property.

To use the COV system, users must enable grouped readings. To do so, configure the **Disable Read Grouping and COV** as False.

After enabling this system, this Driver starts communicating with a device and checks whether that device supports a COV system. If not, this Driver uses grouped or simple readings.

The properties that support COV are **presentValue** and **statusFlags** for objects **analogInput**, **analogOutput**, **analogValue**, **binaryInput**, **binaryOutput**, **binaryValue**, **lifeSafetyPoint**, **lifeSafetyZone**, **multistateInput**, **multistateOutput**, and **multistateValue** and properties **presentValue**, **statusFlags**, **controlledVariableValue**, and **setpoint** for the **loop** object.

To use COV subscriptions, it is recommended that this Driver listens to a fixed port, which is used by a device to send notifications. If the listening port is variable, such as when connecting to a fixed port without specifying a local port, each execution creates new subscriptions, and therefore this Driver cannot remove old subscriptions still active.

Before creating a COV subscription, this Driver checks the existence of a property and then establishes an initial value for the Tag, because COV subscriptions only send values when a property changes.

Notice that in UDP protocol COV notifications sent by a device may be lost, possibly leaving Tags with frozen values. When identifying frozen values, a possible cause could be discarded UDP packets by the network. In this case, disable the COV subscription and return to reading by polling.

Tag Reference

The BACnet Driver implements only services for reading and writing object properties, that is, the **ReadProperty** and **WriteProperty** services. This configuration can be performed with *N* parameters as well as with **Strings**, by using the **ParamDevice** and **ParamItem** properties.

N-Addressing Parameters of PLC Tags

N1	Device number
N2	Object type
N3	Object address
N4	Number of the property for reading or writing

Please check **Appendix I** for a list of objects and properties supported by this Driver.

Configuration by Strings

To use a configuration by **Strings**, the syntax is the following:

```
ParamDevice - "<device#>:[<net>[.<mac>]][,seg]"
```

Where:

- **<device>**: Number of a BACnet device
- **<net>**: Number of a sub-network, optional
- **<mac>**: MAC address inside the sub-network, optional. It must be a sequence of hexadecimal characters, such as "0D" or "1AB0"
- **<seg>**: Segmentation supported by a device, optional. If not informed, assumes the value 0 (zero). Possible values are **0**: Supports segmentation when sending and when receiving (default), **1**: Supports segmentation only when sending, **2**: Supports segmentation only when receiving, or **3**: Does not support segmentation

Examples:

- "200:" accesses a device with number 200
- "200:21" accesses a device with number 200 on sub-network 21
- "200:21.0D" accesses a device with number 200 on sub-network 21, with a MAC address equal to 0D, in hexadecimal
- "200;3" accesses a device with number 200, which does not support packet segmentation
- "200:21,1" accesses a device numbered 200 on sub-network 21, and this device supports segmentation only when sending packets, that is, it does not accept receiving segmented packets

```
ParamItem - "[<deviceAddr><type>-<addr>[.<property>][ P<priority>][ <special>]"
```

Where:

- **<deviceAddr>**: Device address, optional with the same syntax of **ParamDevice**
- **<type>**: Name of an object type, such as "analogInput" or "analogOutput"
- **<addr>**: Object address
- **<property>**: Name of an object property, optional. If not informed, assumes the value of the **presentValue** property

- **<priority>**: Default priority used when writing to this Tag, optional. If not informed, assumes the value of a device's default priority
- **<special>**: Text identifying a **Special Tag** function, optional. If not informed, the Tag performs simple read and write operations

Examples:

- "device-200.localTime" accesses the **localTime** property of object "device-200"
- "analogInput-12" accesses the current value, the **presentValue** property, of analog input 12
- "analogInput-12 P3" accesses the current value, the **presentValue** property, of analog input 12, whose writing, when performed, uses priority 3 (three)
- "200:binaryValue-25.activeText" accesses the **activeText** property of object "binaryValue-25" on a device with number 200. In this case, **ParamDevice** must be empty

Please check **Appendix I** for a list of objects and properties supported by this Driver.

Examples of Tag Configuration

To read or write the **presentValue** property of object "analogValue-300", on a device with address 15, declare a Tag with the following parameters:

- **N1**: 15 (device with address 15)
- **N2**: 2 (**analogValue**-type object)
- **N3**: 300 (object address)
- **N4**: 85 (the **presentValue** property)

For the same configuration by **Strings**, declare a Tag with the following properties:

- **ParamDevice**: "15:"
- **ParamItem**: "analogValue-300"

Or else:

- **ParamDevice**: "" (an empty **String**)
- **ParamItem**: "15:analogValue-300"

NOTE

Configuration by **Strings** assumes the **presentValue** property if not informed. The value "analogValue-300" is equivalent to "analogValue-300.presentValue".

To read a vendor's name, that is, the **vendorName** property from a device with number 15, declare a Tag with the following parameters:

- **N1**: 15 (device with address 15)

- **N2:** 8 (**device**-type object)
- **N3:** 15 (address of object "device-15")
- **N4:** 121 (the **vendorName** property)

For the same configuration by **Strings**, declare a Tag with the following properties:

- **ParamDevice:** "15:"
- **ParamItem:** "device-15.vendorName"

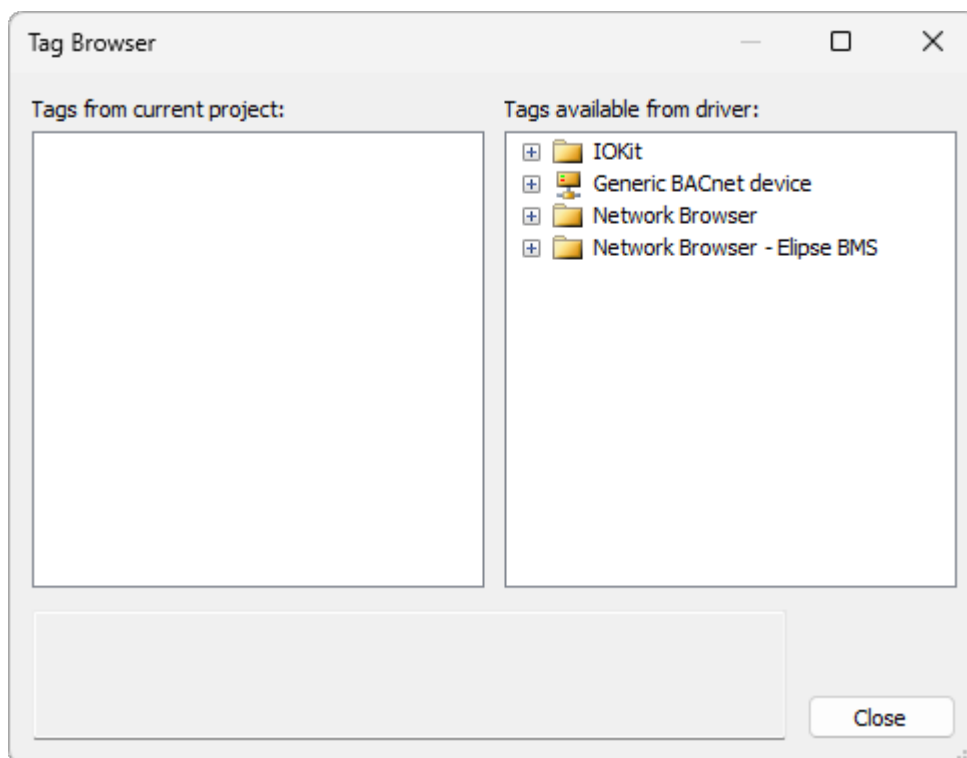
Or else:

- **ParamDevice:** "" (an empty **String**)
- **ParamItem:** "15:device-15.vendorName"

Using Tag Browser

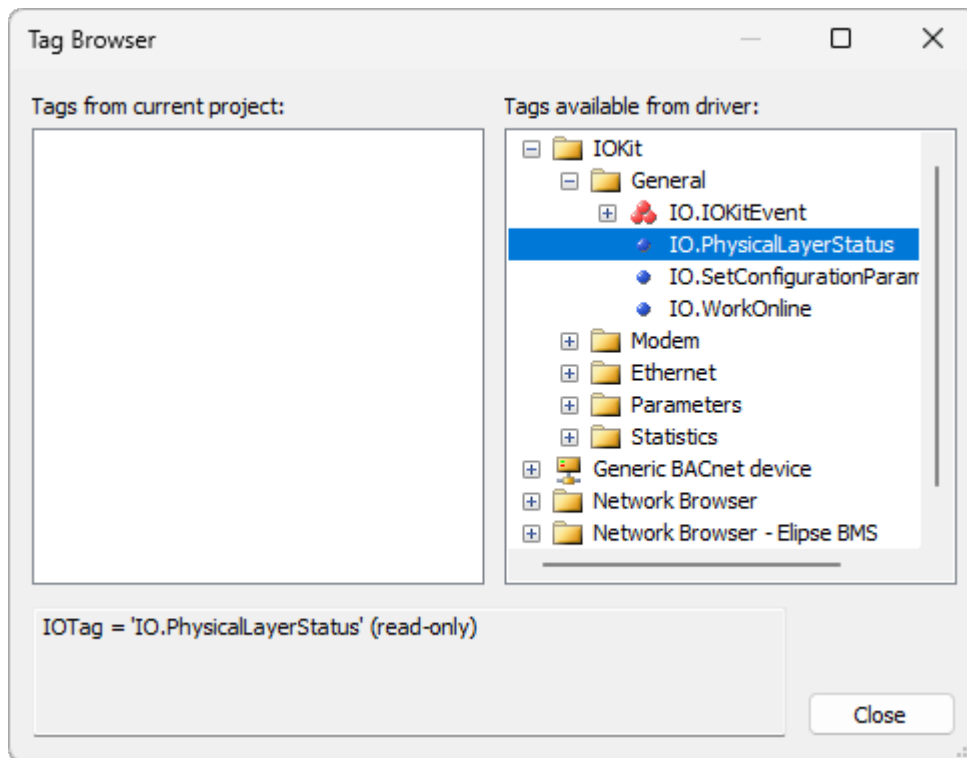
Eclipse E3 version 2.0 or later contains a tool called **Tag Browser**, which helps users to create and configure Tags.

The **Tags from current project** list displays existing Tags and folders in the current project. The **Tags available from driver** list displays a tree with Tags available on BACnet Driver. To create a new Tag in an application, drag one of the Tags defined by this Driver to a folder on the current directory. BACnet Driver contains the nodes displayed on the next figure.



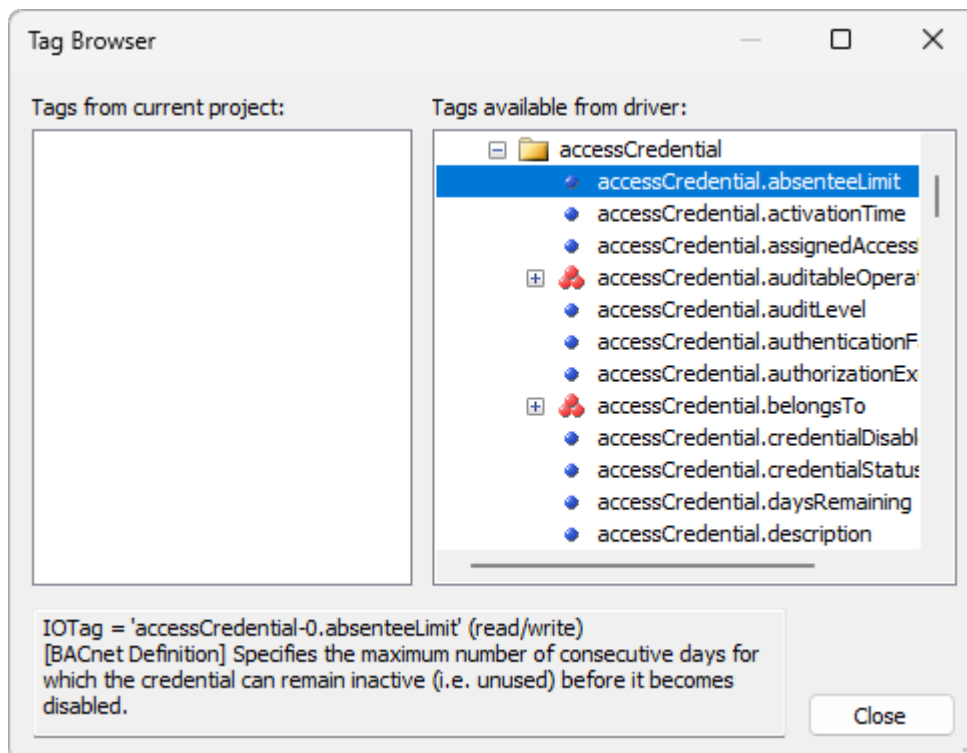
Tag Browser window

The **IOKit** node contains Tags from the **IOKit** library, divided into categories **General**: Tags for general use, **Modem**: Tags to manipulate communication via modem, **Ethernet**: Tags to manipulate communication via Ethernet network, **Parameters**: Tags to configure **IOKit** parameters, and **Statistics**: Tags with statistical information about I/O Interfaces.



Tags from IOKit

The **Generic BACnet device** node lists all types of objects and properties defined by the BACnet protocol. After dragging a generic Tag to the **Tags from current project** list, users must fill in the syntactic parameter **Device** with a device's instance. The syntactic parameter **Item** is already configured to the corresponding object of instance 0 (zero). To access another object of the same type, users must update the object identifier. Please check topic **Configuration by Strings** for more information.

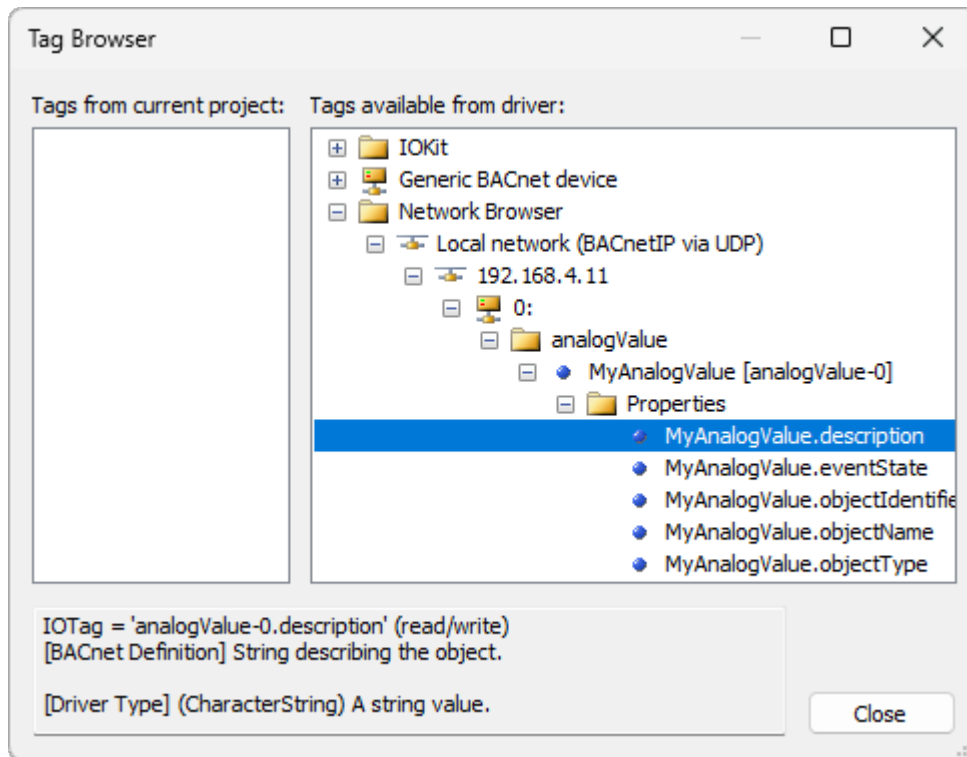


Generic BACnet Device node

The **Network Browser** node allows accessing BACnet devices available on the local network. After expanding this item, nodes relative to the physical addresses are displayed. By expanding these nodes, devices registered in the devices list from the **BACnet** tab are displayed. If this list is empty, this Driver starts **Auto-Detecting Devices** services to search for BACnet devices on the local network and returns a list of all devices found.

After expanding a device node, this Driver reads the **objectList** property of a given **Device** object. This Driver then provides a set of folders, each for an object type, where Tags for all existing objects of a given type can be found. Each Tag name is formed by the values of the properties **objectName**, read by this Driver, and the **objectIdentifier**, obtained from the object list, in the format **objectName [objectIdentifier]**. This Tag accesses the **presentValue** property of this object, if it exists.

After expanding an object's Tag, this Driver reads the **propertyList** of that object. This Driver then provides a **Properties** folder, where Tags for each existing property of that object can be obtained.



Network Browser node

In case any communication errors occur while browsing Tag Browser, these errors are displayed as nodes within Tag Browser.

In a similar, although more simplified way, the **Network Browser - Elipse BMS** node also allows users to access the BACnet devices available on the local network, but rather uploading the most interesting properties to the BMS (*Building Management System*) context.

Special Tags

Some BACnet functionality is achieved through the special Tags described on the next topics.

These special Tags must be configured using the syntactic parameter **Item**. Some of these Tags must also define a specific device using the syntactic parameter **Device**.

AcknowledgeAlarm

The **AcknowledgeAlarm** service sends a request to acknowledge a pending alarm on a device. This Tag must be configured as **Write-Only**. Writing an array with the format specified on the next table sends this service. To inform the destination device when writing a value, the array must have 6 (six) elements. If the device is already configured through the **Item** parameter, the array must have 5 (five) elements.

Values and formats for writing to the AcknowledgeAlarm Tag Array

FIELD	INDEX (6-ELEMENT ARRAY)	INDEX (5-ELEMENT ARRAY)	FORMAT AND DESCRIPTION
Device Identifier	0	-	Identifier of the destination device
Process Identifier	1	0	Identifier of the process acknowledging the alarm
Event Object Identifier	2	1	Text in the format <object>-<identifier> informing the object that triggered the alarm to acknowledge. If the object's type is Event Enrollment , then only the object's identifier can be informed
Event State Acknowledged	3	2	Value representing the state of the alarm to acknowledge. For more information, please check topic BACnetEventState
Alarm Timestamp	4	3	Date and time with the timestamp of the alarm to acknowledge
Acknowledgment Source	5	4	Text identifying who is acknowledging the alarm

ChangeWritePriority

Allows changing the default value of the writing priority of a device. If the **Device** field is addressed, this Tag must inform only the new priority value, between 0 (zero) and 16. If the **Device** field is not addressed, this Tag must be an array in which the first field is the device's address and the second field is the priority.

GetAlarmSummary

The **GetAlarmSummary** service is used to retrieve a list of active alarms in a device. The execution of this service is implemented using two Tags. The **GetAlarmSummaryCommand** Tag is used to trigger this service and the **GetAlarmSummaryResult** Tag receives the result of this operation. Both Tags use a **Configuration by Strings** to reference a specific device.

The **GetAlarmSummaryCommand** must be configured as a write-only Tag. To start this service, write any value to this Tag.

The **GetAlarmSummaryResult** must be a read-only Block Tag with 3 (three) Elements, which receives a list of results. The Elements and data types are the following:

- **Element 0:** *Object Identifier*, a text value in the format **<object>-<identifier>**
- **Element 1:** *Alarm State*, an unsigned integer value representing the **BACnetEventState** enumeration
- **Element 2:** *Acknowledged Transitions*, a text value representing a sequence of bits

GetEnrollmentSummary

The **GetEnrollmentSummary** service is used to retrieve a list of objects that start events of a device. This Driver sends this service with the following filters:

- **Acknowledgment Filter:** all
- **Event State Filter:** active

The execution of this service is implemented using two Tags. The **GetEnrollmentSummaryCommand** Tag is used to trigger this service and the **GetEnrollmentSummaryResult** Tag receives the result of this operation. Both Tags use a **Configuration by Strings** to reference a specific device.

The **GetEnrollmentSummaryCommand** must be a write-only Tag. To start this service, write any value to this Tag.

The **GetEnrollmentSummaryResult** must be a read-only Block Tag with 5 (five) Elements, which receives a list of results. The Elements and data types are the following:

- **Element 0:** *Object Identifier*, a text value in the format **<object>-<identifier>**
- **Element 1:** *Event Type*, an unsigned integer value representing the **BACnetEventType** enumeration
- **Element 2:** *Event State*, an unsigned integer value representing the **BACnetEventState** enumeration
- **Element 3:** *Priority*, an unsigned integer value
- **Element 4:** *Notification Class*, an unsigned integer value

GetEventInformation

The **GetEventInformation** service is used to retrieve a list of all active events of a device. The execution of this service is implemented using two Tags. The **GetEventInformationCommand** Tag is used to trigger this service and the **GetEventInformationResult** Tag receives the result of this operation. Both Tags use a **Configuration by Strings** to reference a specific device.

This Driver automatically manages sending new requests if the events do not fit a single response. In this case, the result is provided when all events are received.

The **GetEventInformationCommand** must be a write-only Tag. To start this service, write any value to this Tag.

The **GetEventInformationResult** must be a read-only Block Tag with 7 (seven) Elements, which receives a list of results. The Elements and data types are the following:

- **Element 0:** *Object Identifier*, a text value in the format **<object>-<identifier>**
- **Element 1:** *Event State*, an unsigned integer value representing the **BACnetEventState** enumeration
- **Element 2:** *Acknowledged Transitions*, a text value representing a sequence of bits
- **Element 3:** *Event Timestamps*, a text value in **XML** format
- **Element 4:** *Notify Type*, an unsigned integer value representing the **BACnetNotifyType** enumeration
- **Element 5:** *Event Enable*, a text value representing a sequence of bits
- **Element 6:** *Event Priorities*, a text value in **XML** format

ReadRange

The **ReadRange** service reads a range of data from a **List**-type property. The execution of this service is implemented using two Tags. The **ReadRangeCommand** Tag is used to trigger this service and the **ReadRangeResult** Tag receives the result of this operation.

Both Tags use a **Configuration by Strings** to reference a specific device, object, and property. The referenced property must have a **List** type. At the end of the **Item** parameter, separated by a space, a text must be included identifying the Tag, as in the next example.

- **ParamDevice:** "15:"
- **ParamItem:** "trendLog-10.logBuffer ReadRangeCommand"

The **ReadRangeCommand** must be a write-only Tag. To start this service, this Tag must receive the writing of an array with 3 (three) Elements, in the following format:

- **Element 0:** *Type*, a text value identifying the type of range to read. Valid values are **FullRange**, **Position**, **SequenceNumber**, or **Time**
- **Element 1:** *Reference*, a reference to the initial or final Element. If the type of range is equal to **Position** or **SequenceNumber**, this must be an unsigned integer. If the type of range is equal to **Time**, this must be a date and time
- **Element 2:** *Count*, an integer value informing the number of Elements to read. For positive values, the reference is the beginning of the range. For negative values, the reference is the end of the range

The **ReadRangeResult** Tag must be configured as a read-only Block Tag with 6 (six) Elements. The Elements and their data types are the following:

- **Element 0:** *First Item*, a Boolean value
- **Element 1:** *Last Item*, a Boolean value
- **Element 2:** *More Items*, a Boolean value
- **Element 3:** *Item Count*, an unsigned integer value
- **Element 4:** *Item Data*, a text value in **XML** format
- **Element 5:** *First Sequence Number*, an unsigned integer value, available only in **SequenceNumber**- and **Time**-type ranges

ReceiveEventNotifications

This read-only Block Tag presents notifications received using the **ConfirmedEventNotification** and **UnconfirmedEventNotification** services, as well as the alarm summary read during this Driver's initialization. This Block Tag must have 13 Elements, described next.

- **Element 0:** *Process Identifier*, an unsigned integer value
- **Element 1:** *Initiating Device Identifier*, an unsigned integer value
- **Element 2:** *Event Object Identifier*, a text value in the format **<object>-<identifier>**
- **Element 3:** *Timestamp*, a date and time value
- **Element 4:** *Notification Class*, an unsigned integer value

- **Element 5:** *Priority*, an unsigned integer value
- **Element 6:** *Event Type*, an unsigned integer value representing the **BACnetEventType** enumeration
- **Element 7:** *Message Text*, a text value
- **Element 8:** *Notify Type*, an unsigned integer value representing the **BACnetNotifyType** enumeration
- **Element 9:** *Ack Required*, a Boolean value
- **Element 10:** *From State*, an unsigned integer value representing the **BACnetEventState** enumeration
- **Element 11:** *To State*, an unsigned integer value representing the **BACnetEventState** enumeration
- **Element 12:** *Event Values*, a text value in **XML** format

RuntimeBrowseRead

Receives the result of a browsing operation on the network, as requested by the **RuntimeBrowseWrite** Tag. In case of performing a request for names of devices or names of objects from a device, this reading returns a list of names, that is, an array that can be handled by a Simple Tag or a Block Tag with one or more Elements, although it only contains values for its first Element. In case of performing a request for properties of an object, this reading returns a two-column matrix, which must be handled by a Block Tag with two Elements, where the first Element corresponds to the property's name and the second Element corresponds to the property's current value.

RuntimeBrowseWrite

Requests the execution of a browsing operation on the network to retrieve devices and their variables. The result of this browsing operation is received in the **RuntimeBrowseRead** Tag. This Tag must receive, when writing, a context name in **String** format. If that context is empty, then the browsing operation searches for device names. If that context is the name of a device, the browsing operation searches for the name of its objects. If that context is the name of an object, the browsing operation searches for the name of its properties.

TimeSynchronization and UTCTimeSynchronization

The **TimeSynchronization** and **UTCTimeSynchronization** services are used by this Driver to notify other devices of the current time, retrieved from the operating system. The **TimeSynchronization** service sends the time based on the local timezone, while the **UTCTimeSynchronization** service sends the time based on the UTC (*Coordinated Universal Time*) timezone.

These Tags must be configured as **Write-Only**. Writing any value to these Tags triggers the sending service. When these Tags specify a device, the service is sent only to that device. When no device is specified, the service is sent in broadcast mode.

WritePropertyMultiple

The **WritePropertyMultiple** service is used to write values to multiple properties of multiple objects in a single device by using a single message. The execution of this service is implemented using special Tags. The **WritePropertyMultipleAdd** Tag adds a value to the list of pending writings, the **WritePropertyMultipleCommand** Tag is used to trigger this service, the **WritePropertyMultiplePending** Tag shows the list of pending writings, and the **WritePropertyMultipleResult** Tag shows the result of the last multiple writing.

This Driver can store up to 20 values on its list of pending writings. This list is sorted by the type of the object, then by the instance of the object, and then by the property. After sending a multiple writing service, the values whose writing were successful are removed from that list.

All these Tags use a **Configuration by Strings**. The **WritePropertyMultipleAdd** Tag must reference a specific device, object, and property, and also an optional priority. All other Tags must reference only a specific device. At the end of the **Item** parameter, separated by a space, there must be a text identifying the Tag, such as in the next example:

- **ParamDevice:** "15:"
- **ParamItem:** "analogInput-12.presentValue P3 WritePropertyMultipleAdd"

Or

- **ParamDevice:** "15:"
- **ParamItem:** "WritePropertyMultipleCommand"

The **WritePropertyMultipleAdd** Tag must be configured as a write-only Tag. This Tag can be Simple or Block, depending on the format linked to the property to write. For more information, please check **Appendix I**. When writing a value to this Tag, it is stored on a list of pending writings. If that list is full, the value is discarded. If a new value is written before sending the previous value, that previous value is discarded.

The **WritePropertyMultipleCommand** Tag must be configured as a write-only Tag. This Tag performs the following actions, depending on the value written to it:

- **Command:** Sends a **WritePropertyMultiple** service with the values stored on the list of pending writings
- **Clear All:** Clears the list of pending writings, discarding all values
- **Clear <object type>-<object id>.<property>:** Clears a specific entry from the list of pending values, referring to an object of type <object type>, an <object id> instance, and a <property> property, discarding its value

The **WritePropertyMultiplePending** Tag must be configured as a read-only Block Tag. The number of Block Elements must be two times the number of values stored on the list of pending writings. Each pair of Block Elements corresponds, successively, to a property identifier and to the value of a pending writing in this property.

The **WritePropertyMultipleResult** Tag must be configured as a read-only Block Tag with two Elements, described next:

- **Element 0:** *Status*, a text informing the result of the last multiple writing. Possible values are the following:
 - **Not Executed:** No multiple writing service has been sent yet
 - **Success:** All writings from the last service sent were successfully executed
 - **Failed:** All writings from the last service sent failed
 - **Error(<error class>.<error code>):** Writings from the last service sent were partially successful and interrupted by the specified error
- **Element 1:** *First Failed Write Attempt*, a text identifying the property from the last service sent whose writing resulted in an error. All properties after that on the list of pending writings were not executed. This Element is only filled in case of partial errors

Documentation of I/O Interfaces

This section contains the documentation of I/O Interfaces referring to **BACnet** Driver.

Configuration of a Driver

I/O Interface configuration is performed on a Driver's configuration dialog box. To access the configuration of this dialog box in **Eclipse E3** in version 1.0, follow these steps:

1. Right-click a Driver object (IODriver).
2. Select the **Properties** item on the contextual menu.
3. Select the **Driver** tab.
4. Click **Other parameters**.

In **Eclipse E3** version 2.0 or later, click **Configure driver**  on a Driver's toolbar. In **Eclipse SCADA**, follow these steps:

1. Open the Organizer.
2. Select a Driver on Organizer's tree.
3. Click **Extras** on the **Driver** tab.

Currently, an I/O Interface allows opening only one connection for each Driver. This means that, if users want to access two serial ports, they must add two Drivers to an application and then configure each one of these Drivers for each serial port.

Configuration Dialog Box

The dialog box of I/O Interfaces allows configuring the I/O connection used by a Driver. This dialog box contains the **Setup**, **Serial**, **Ethernet**, **Modem**, and **RAS** tabs, described on the next topics. If a Driver does not implement a specific I/O connection, its corresponding tab is not available for configuration. Some Drivers may contain additional tabs, specific for that Driver, on the configuration dialog box.

Setup Tab

The **Setup** tab contains general configurations of a Driver. This tab is divided into the following groups:

- **General configurations:** Configurations of a Driver's physical layer, time-out, and initialization mode
- **Connection management:** Configurations on how the I/O Interface keeps a connection and which recovery policy is used on failure
- **Logging options:** Controls the generation of log files

Setup

Physical Layer: Ethernet ☐ Start driver OFFLINE

Timeout: 1000 ms Communication check time: 5000 ms

Connection management

Mode: Automatic (managed by the driver)

☒ Retry failed connection every 20 seconds

☐ Give up after 1 failed retries

☐ Disconnect if non-responsive for 0 seconds

Logging Options

☐ Log to File: C:\eeLogs\MicrolokII_%DATE%.log

File size limit (MB): 0 ('0' is unlimited)

Setup tab

General options on the Setup tab

OPTION	DESCRIPTION
Physical Layer	Select the physical layer on a list. Available options are Serial , Ethernet , Modem , and RAS . The selected interface must be configured on its specific tab
Timeout	Configure a time-out, in milliseconds, for the physical layer. This is the amount of time an I/O interface waits to receive any byte from the reception's buffer
Communication check time	Set the time, in milliseconds, to define the interval at which communication is considered to be in an inactive state. As long as an I/O Driver receives valid data, its communication state is considered active. However, if during operation an I/O Driver does not receive valid data inside this period of time, the state is considered inactive. The communication state is shown in the IO.CommunicationStatus Tag
Start driver OFFLINE	Select this option so that a Driver starts in Offline mode or stopped. This means that the I/O interface is not created until this Driver is configured to Online mode by using a Tag in an application. This mode enables a dynamic configuration of an I/O interface at run time

Options on the Connection management group

OPTION	DESCRIPTION
Mode	Selects a management mode of a connection. Selecting the Automatic option allows a Driver to manage the connection automatically, as specified in the next options. Selecting the Manual option allows an application to fully manage a connection. Please check topic Driver Statuses for more information
Retry failed connection every ... seconds	Select this option to enable a Driver's connection retry in a certain interval, in seconds. If the Give up after failed retries option is not selected, this Driver keeps retrying until a connection is performed, or until the application is stopped
Give up after ... failed retries	Enable this option to define a maximum number of connection retries. When the specified number of consecutive connection retries is reached, a Driver goes to the Offline mode, assuming that a hardware problem was detected. If a Driver establishes a successful connection, the number of unsuccessful retries is cleared. If this new connection is lost, then the retry counter starts at zero
Disconnect if non-responsive for ... seconds	Enable this option to force a Driver to disconnect if no byte was received by the I/O interface during the specified time-out, in seconds. This time-out must be greater than the time-out configured in the Timeout option

Options on the Logging Options group

OPTION	DESCRIPTION
Log to File	Enable this option and configure the name of a file to write a log. Log files can be large, so use this option for short periods of time, only for testing and debugging purposes. If the %PROCESS% macro is used in the log file name, it is replaced by the identifier of the current process. This option is particularly useful when using several instances of the same Driver in Eclipse E3, thus allowing each instance to generate a separate log file. For example, when configuring this option with value "c:\e3logs\drivers\sim_%PROCESS%.log", it generates a file named c:\e3logs\drivers\sim_00000FDA.log for process 0FDAh. Users can also use the %DATE% macro in the file name. In this case a log file is generated every day, in the format aaaa_mm_dd. For example, when configuring this option with value "c:\e3logs\drivers\sim_%DATE%.log", it generates a file named c:\e3logs\drivers\sim_2005_12_31.log in 12/31/2005 and a file named c:\e3logs\drivers\sim_2006_01_01.log in 01/01/2006. Similarly, the %DATE_HOUR% macro generates one log file per hour, in the format aaaa_mm_dd_hh
File size limit (MB)	Configure the log file size limit, in megabytes. A value equal to 0 (zero) means that there is no size limit for the log file

Serial Tab

Use this tab to configure parameters for a **Serial** Interface.

Serial

Port: COM1

Baud rate: 9600

Data bits: 8 data bits

Parity: None

Stop bits: 1 stop bit

☐ Enable 'ECHO' suppression

Handshaking

DTR control: OFF

RTS control: OFF

☐ Wait for CTS before send

CTS timeout: 0 ms

Delay before send: 0 ms

Delay after send: 0 ms

Inter-byte delay (microseconds): 0 μ s

Inter-frame delay (milliseconds): 0 ms

Serial tab

General options on the Serial tab

OPTION	DESCRIPTION
Port	Select a serial port on the list, from COM1 to COM4 , or type the name of a serial port in the format COMn , such as "COM15". When typing the name of a serial port manually, the dialog box only accepts names of serial ports starting with the expression "COM"
Baud rate	Select a baud rate on the list (1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200) or type a baud rate, such as 600
Data bits	Select 7 (seven) or 8 (eight) data bits on the list
Parity	Select a parity on the list. The available options are None, Even, Odd, Mark, or List
Stop bits	Select the number of stop bits on the list. The available options are 1, 1.5, or 2 stop bits
Enable 'ECHO' suppression	Enable this option to remove the echo received after the I/O Interface sends data via serial port. If this echo is not equal to the bytes just sent, then the I/O Interface aborts communication
Inter-byte delay (microseconds)	Defines a delay between each byte transmitted by the I/O Interface, in millionths of a second, that is, 1000000 is equal to a second. This option must be used with small delays of less than a millisecond
Inter-frame delay (milliseconds)	Defines a delay between packets sent or received by the I/O Interface, in thousandths of a second, that is, 1000 is equal to a second. This delay is applied if the I/O Interface

OPTION	DESCRIPTION
	sends two consecutive packets, or between a received packet and the next sending

The **Handshaking** group configures the usage of **RTS**, **CTS**, and **DTR** signals in the handshaking process, that is, it controls when data can be sent or received via serial line. Most of the time, configuring the **DTR control** option to **ON** and the **RTS control** option to **Toggle** works with **RS232**-type serial lines as well as with **RS485**-type serial lines.

Available options on the Handshaking group

OPTION	DESCRIPTION
DTR control	Select the value ON to keep the DTR signal always on while the serial port is open. Select the value OFF to turn the DTR signal off while the serial port is open. Some devices require the DTR signal always on to allow communication
RTS control	Select the value ON to keep the RTS signal always on while the serial port is open. Select the value OFF to turn the RTS signal off while the serial port is open. Select the value Toggle to turn the RTS signal on while sending bytes via serial port and turn it off when not sending bytes, therefore enabling the reception
Wait for CTS before send	Available only when the RTS control option is configured with the value Toggle . Use this option to force a Driver to check the CTS signal before sending bytes via serial port, after turning the RTS signal on. In this mode, the CTS signal is handled as a permission flag for sending
CTS timeout	Determines a maximum time, in milliseconds, that a Driver waits for the CTS signal after turning the RTS signal on. If the CTS signal is not turned on within this time-out, that Driver then fails the current communication and returns an error
Delay before send	Some serial port devices have a delay when enabling a data sending circuit after the RTS signal is turned on. Configure this option to wait a certain number of milliseconds after turning the RTS signal on and before sending the first byte. IMPORTANT: This delay must be used carefully, because it uses 100% of CPU resources while waiting. System's general performance degrades as this value increases
Delay after send	This is the same effect of the Delay before send option, but in this case the delay is performed after sending the last byte, before turning the RTS signal off

Ethernet Tab

Use this tab to configure parameters of an **Ethernet** Interface. These parameters, except port configurations, must also be configured for use in the **RAS** Interface.

Ethernet

Transport: TCP/IP ☐ Listen for connections on port: 0

☐ PING before connecting ☐ Share listen port with other processes

Timeout: 4000 ms ☐ Interface: (All Interfaces)

Retries: 1 ☐ Use IPv6 ☐ Use SSL SSL Settings

☐ Enable 'ECHO' suppression

IP Filter:

Connect to

Main IP: Port: 502 ☐ Local port: 0

☐ Backup IP 1: Port: 0 ☐ Local port: 0

☐ Backup IP 2: Port: 0 ☐ Local port: 0

☐ Backup IP 3: Port: 0 ☐ Local port: 0

Ethernet tab**Available options on the Ethernet tab**

OPTION	DESCRIPTION
Transport	Select the value TCP/IP for a TCP socket (<i>stream</i>) or select the value UDP/IP to use a UDP socket (<i>connectionless datagram</i>)
Listen for connections on port	Use this option to wait for new connections in a specific IP port, common in Slave Drivers. If this option remains unselected, a Driver connects to the address and port specified in the Connect to option
Share listen port with other processes	Select this option to share the listening port with other Drivers and processes
Interface	Select the local network interface, identified by its IP address, that a Driver uses to establish and receive connections, or select the value (All Interfaces) to allow connection in any network interface
Use IPv6	Select this option to force a Driver to use addresses in IPv6 format on all Ethernet connections. Leave this option deselected to use the IPv4 format
Enable 'ECHO' suppression	Enable this option to remove the echo from received data. An echo is a copy of sent data, which can be returned before a reply message
IP Filter	List of restricted or allowed IP addresses from where a Driver accepts connections (<i>Firewall</i>). Please check the IO.Ethernet.IPFilter property for more information
PING before connecting	Enable this option to execute a ping command, that is, to check whether a device can be reached on a network, for a device before trying a socket connection. This is a quick way

OPTION	DESCRIPTION
	of determining a successful connection before trying to open a socket with a device. The time-out of a connection with a socket can be very high. The available options are: • Timeout: Specify the number of milliseconds to wait for a reply from a ping command. Users must use a ping command to check the normal reply time, configuring this option for a value above that average. Usually this value can be configured between 1000 and 4000 milliseconds, that is, between 1 (one) and 4 (four) seconds • Retries: Number of retries of a ping command, not counting the first attempt. If all attempts fail, then the socket connection is aborted

Available options on the Connect to group

OPTION	DESCRIPTION
Main IP	Type the IP address of a remote device. Users can use an IP address separated by dots, as well as a URL. In case of a URL, a Driver uses the available DNS service to map that URL to an IP address, such as "192.168.0.13" or "Server1"
Port	Type the IP port of a remote device, between 0 (zero) and 65535
Local port	Select this option to use a fixed local IP port when connecting to a remote device
Backup IP 1, 2, and 3	Indicate the IP address, the IP port, and the fixed local IP port of up to 3 (three) backup addresses of a remote device

Modem Tab

Use this tab to configure parameters of a **Modem** Interface. Some options on the **Serial** tab affect the configuration of a modem, therefore users must also configure the **Serial** Interface.

Modem

Select the modem to use:

Modem settings...

Dial Number:

☐ Accept incoming calls

Modem tab

The **Modem** Interface uses the TAPI modems installed on the computer.

Available options on the Modem tab

OPTION	DESCRIPTION
Select the modem to use	Select a modem on the list of available modems on the computer. If the value Default modem is selected, then the first available modem is used. Selecting this option is recommended specially when an application is used on another computer
Modem settings	Click to open the configuration window of the selected modem
Dial Number	Type a default number for dialing. This value can be changed at run time. Users can use the w character to represent a pause or a waiting time for a dial tone. For example, "0w33313456" dials the number 0 (zero), waits, and then dials the number "33313456"
Accept incoming calls	Enable this option so that a Driver answers the phone when receiving an external call. To use this option, users must configure the Connection management option on the Setup tab to the value Manual

RAS Tab

Use this tab configure parameters of a **RAS** Interface. Users must also configure the **Ethernet** tab.

A **RAS** Interface opens a socket connection with a RAS device. A RAS device is a server of modems available through TCP/IP, waiting for socket connections on an IP port. For each connection accepted on this port, users have access to one modem.

When connecting to a RAS device, first the I/O Interface**IOKit** connects to the socket on the IP address and port configured on the **Ethernet** tab. After opening the socket, the following initialization or connection steps are performed:

1. Clears the socket, that is, removes any **TELNET** greeting message received from a RAS device.
2. Sends an **AT** dial message, in **ASCII** format, in the socket.
3. Waits for a **CONNECT** reply.
4. If the time-out expires, the connection is aborted.
5. If the **CONNECT** reply is received within the time-out, the socket is available for communication with a device, that is, the connection was established.

If step 5 (five) is successful, then the socket behaves as a normal socket, with the RAS device working as a router between a Driver and the device. Bytes sent by a Driver are received by the RAS device and sent to the destination device using a modem. Bytes received by the modem's RAS device are sent back to a Driver using the same socket.

After establishing a connection, the **RAS** interface monitors data received by a Driver. If a "NO CARRIER" **String** is found, the socket is closed. If the RAS device does not send a **NO CARRIER** signal, the **RAS** Interface cannot detect when the modem connection between the RAS device and the final I/O device fails. To recover from this failure, users are strongly advised to enable the **Disconnect if non-responsive** option on the **Setup** tab.

RAS

AT command:

Connection timeout:

0

seconds

Other socket settings should be configured in the "Ethernet" tab!

RAS tab

Available options on RAS tab

OPTION	DESCRIPTION
AT command	A String with the full AT command used to dial to a destination device. For example, "ATDT33313456" dials by tone to number "33313456"
Connection timeout	Number of seconds to wait for a modem's CONNECT reply, after sending an AT command

General Configurations

This section contains information about the configuration of general **I/O Tags** and **Properties** of I/O Interfaces.

I/O Tags

General I/O Interfaces Tags (N2/B2 = 0)

The Tags described next are provided for all supported I/O Interfaces.

IO.CommunicationStatus

Type of Tag	I/O Tag
Type of Access	Reading
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	6 (six)
String Configuration	IO.CommunicationStatus

This Tag informs the communication status of a Driver. It indicates how communication works relative to receiving valid data within a time period arbitrated in the configuration. For more information, please check topic **Setup Tab**. Possible values are **0 - Inactive communication**: The Driver did not receive valid data or stopped receiving data after *n* milliseconds, as configured in the properties window, or **1 - Active communication**: The Driver is receiving valid data.

IO.IOKitEvent

Type of Tag	Block Tag
Type of Access	Read-Only
B1 Parameter	-1 (minus one)
B2 Parameter	0 (zero)
B3 Parameter	0 (zero)
B4 Parameter	1 (one)
Size Property	4 (four)
ParamItem Property	IO.IOKitEvent

This Block returns Driver events generated by several sources in I/O Interfaces. The **TimeStamp** property of this Block represents the moment this event occurred. The Block Elements are the following:

- **Element 0**: Type of event. Possible values are **0**: Information, **1**: Warning, or **2**: Error
- **Element 1**: Source of an event. Possible values are **0**: Driver (specific of a Driver), **-1**: IOKit (generic events of I/O Interfaces), **-2**: **Serial** Interface, **-3**: **Modem** Interface, **-4**: **Ethernet** Interface, or **-5**: **RAS** Interface
- **Element 2**: Error number, specific for each source of event

- **Element 3:** Message of an event, a **String** specific for each event

NOTE

A Driver keeps a maximum number of 100 events internally. If additional events are reported, older events are discarded.

IO.PhysicalLayerStatus

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	2 (two)
String Configuration	IO.PhysicalLayerStatus

This Tag indicates the status of a physical layer. Possible values are the following:

- **0:** Physical layer stopped, that is, a Driver is in **Offline** mode, the physical layer failed when initializing, or exceeded the maximum number of reconnection attempts
- **1:** Physical layer started but not connected, that is, a Driver is in **Online** mode but the physical layer is not connected. If the **Connection management** option is configured with the value **Automatic**, the physical layer can be connecting, disconnecting, or waiting for a reconnection attempt. If the **Connection management** option is configured with the value **Manual**, then the physical layer remains in this status until forced to connect
- **2:** Physical layer connected, that is, the physical layer is ready for use. This **DOES NOT** mean a device is connected, only that the access layer is working

IO.SetConfigurationParameters

Type of Tag	Block Tag
Type of Access	Read-Only
B1 Parameter	-1 (minus one)
B2 Parameter	0 (zero)
B3 Parameter	0 (zero)
B4 Parameter	3 (three)
Size Property	2 (two)
ParamItem Property	IO.SetConfigurationParameters

Use this Tag to change any property of a Driver's configuration dialog box at run time.

This Tag works only while a Driver is in **Offline** mode. To start a Driver in **Offline** mode, select the **Start driver OFFLINE** option on that Driver's configuration dialog box. Users can write to a PLC Tag or to a Block Tag containing the parameters to change. Writing individual Block Elements is not supported, the whole Block must be written at once.

In **Elipse SCADA**, users must use a Block Tag. Every parameter to configure uses two Block Elements. For example, if users want to configure 3 (three) parameters, then the size of the Block must be 6 (six, 3×2). The first Element is the property's name, as a **String**, and the second Element is the property's value, according to the next example.

```
// 'Block' must be a Block Tag with automatic reading,
// scan reading, and automatic writings disabled.
// Configure all parameters
Block.element001 = "IO.Type" // Parameter 1
Block.element002 = "Serial"
Block.element003 = "IO.Serial.Port" // Parameter 2
Block.element004 = 1
Block.element005 = "IO.Serial.BaudRate" // Parameter 3
Block.element006 = 19200
// Writes the whole Block
Block.Write()
```

When using **Elipse E3**, the ability to create arrays at run time allows using an I/O Tag as well as a Block Tag. Users can use the **Write** method of a Driver to send the parameters directly to that Driver, without creating a Tag, according to the next example.

```
Dim arr(6)
' Configure all array elements
arr(1) = "IO.Type"
arr(2) = "Serial"
arr(3) = "IO.Serial.Port"
arr(4) = 1
arr(5) = "IO.Serial.BaudRate"
arr(6) = 19200
' There are two methods to send parameters
' Method 1: Using an I/O Tag
tag.WriteEx arr
' Method 2: Without using a Tag
Driver.Write -1, 0, 0, 3, arr
```

A variation of the previous example uses a bidimensional array.

```
Dim arr(10)
' Configure all array elements. Notice the array was resized
' to 10 elements. Empty array elements are ignored by a Driver
arr(1) = Array("IO.Type", "Serial")
arr(2) = Array("IO.Serial.Port", 1)
arr(3) = Array("IO.Serial.BaudRate", 19200)
Driver.Write -1, 0, 0, 3, arr
```

A Driver does not validate parameter names or passed values, therefore be careful when writing parameters and values. The **Write** method fails if the configuration array is incorrectly created. Users can check the log of a Driver or use the **writeStatus** parameter of the **WriteEx** method to find out the exact cause of an error.

```
Dim arr(10), strError
arr(1) = Array("IO.Type", "Serial")
arr(2) = Array("IO.Serial.Port", 1)
arr(3) = Array("IO.Serial.BaudRate", 19200)
If Not Driver.WriteEx -1, 0, 0, 3, arr, , strError Then
    MsgBox "Failed configuring Driver parameters: " + strError
End If
```

IO.WorkOnline

Type of Tag	I/O Tag
Type of Access	Reading or Writing
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	4 (four)
String Configuration	IO.WorkOnline

This Tag informs the current status of a Driver and allows starting or stopping the physical layer. Possible values are the following:

- **0 - Driver Offline:** Physical layer is closed or stopped. This mode allows a dynamic configuration of a Driver's parameters using the **IO.SetConfigurationParameters** Tag
- **1 - Driver Online:** Physical layer is open or executing. While in **Online** mode, the physical layer can be connected or disconnected and its current status can be checked using the **IO.PhysicalLayerStatus** Tag

In the next example, using **Eclipse E3**, a Driver is configured to **Offline** mode, its COM port is changed, and then configured to **Online** mode again.

```
'Configure to Offline mode
Driver.Write -1, 0, 0, 4, 0
'Change port to COM2
Driver.Write -1, 0, 0, 3, Array("IO.Serial.Port", 2)
'Configure to Online mode
Driver.Write -1, 0, 0, 4, 1
```

The **Write** method may fail when configuring a Driver to **Online** mode, that is, writing the value 1 (one). In this case, this Driver remains in **Offline** mode. The cause of failure can be:

- Type of physical layer incorrectly configured, probably an invalid value was configured in the **IO.Type** property
- This Driver may have run out of memory
- Physical layer probably did not create its working thread. Search the log file for a message "Failed to create physical layer thread!"
- Physical layer could not start. The cause of this failure depends on the type of physical layer. It can be an invalid serial port number, a failure when starting Windows Sockets, or a failure when starting TAPI (modem), among others. This cause is recorded on the log file

IMPORTANT

Even if the configuration of a Driver to **Online** mode is successful, this does not necessarily mean the physical layer is ready to use, that is, ready to execute input and output operations with an external device. The **IO.PhysicalLayerStatus** Tag must be checked to ensure the physical layer is connected and ready for communication.

Properties

These are general properties of all supported I/O Interfaces.

IO.ConnectionMode

9 Controls the management mode of a Connection. Possible values are **0**: Automatic mode, in which a Driver manages the connection or **1**: Manual mode, in which an application manages the connection.

IO.GiveUpEnable

☑ When configured to True, defines a maximum number of reconnection attempts. If all reconnection attempts fail, a Driver enters the **Offline** mode. When configured to False, a Driver tries until a reconnection is successful.

IO.GiveUpTries

9 Number of reconnection attempts before this one is aborted. For example, if the value of this property is equal to 1 (one), a Driver tries only one reconnection when the connection is lost. If this one fails, this Driver enters the **Offline** mode.

IO.InactivityEnable

☑ Configure to True to enable and to False to disable inactivity detection. The physical layer is disconnected if inactive for a certain period of time. The physical layer is considered inactive only if it is capable of sending data but not capable of receiving it back.

IO.InactivityPeriodSec

9 Number of seconds to check for inactivity. If the physical layer is inactive for this period of time, it is then disconnected.

IO.RecoverEnable

☑ Configure to True to enable a Driver to recover lost connections and to False to leave a Driver in **Offline** mode when a connection is lost.

IO.RecoverPeriodSec

9 Delay time between two connection attempts, in seconds.

NOTE

The first reconnection is executed immediately after a connection is lost.

IO.StartOffline

☑ Configure to True to start a Driver in **Offline** mode and to False to start a Driver in **Online** mode.

NOTE

It is pointless to change this property at run time, as it can only be changed when a Driver is already in **Offline** mode. To configure a Driver in **Online** mode at run time, write the value 1 (one) to the **IO.WorkOnline** Tag.

IO.TimeoutMs

9 Defines a time-out for the physical layer, in milliseconds. One second is equal to 1000 milliseconds.

IO.Type

 Defines the type of physical interface used by a Driver. Possible values are the following:

- **N or None:** Does not use a physical interface, that is, a Driver must provide a customized interface
- **S or Serial:** Uses a local serial port (COM n)
- **M or Modem:** Uses a local modem, internal or external, accessed via TAPI (*Telephony Application Programming Interface*)
- **E or Ethernet:** Uses a TCP/IP or UDP/IP socket
- **R or RAS:** Uses a **RAS** (*Remote Access Server*) Interface. A Driver connects to a RAS device using the **Ethernet** Interface and then sends an **AT** (*dial*) command

Statistical Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of I/O Interfaces statistics.

I/O Tags

Tags of I/O Interface Statistics (N2/B2 = 0)

The Tags described next display statistics for all I/O Interfaces.

IO.Stats.Partial.BytesRecv

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1101
Configuration by String	IO.Stats.Partial.BytesRecv

This Tag returns the number of bytes received in the current connection.

IO.Stats.Partial.BytesSent

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1100
Configuration by String	IO.Stats.Partial.BytesSent

This Tag returns the number of bytes sent through the current connection.

IO.Stats.Partial.TimeConnectedSeconds

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1102
Configuration by String	IO.Stats.Partial.TimeConnectedSeconds

This Tag returns the number of seconds a Driver is connected in the current connection or 0 (zero) if a Driver is disconnected.

IO.Stats.Partial.TimeDisconnectedSeconds

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1103
Configuration by String	IO.Stats.Partial.TimeDisconnectedSeconds

This Tag returns the number of seconds a Driver is disconnected since the last connection ended or 0 (zero) if a Driver is connected.

IO.Stats.Total.BytesRecv

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1001
Configuration by String	IO.Stats.Total.BytesRecv

This Tag returns the number of bytes received since a Driver was loaded.

IO.Stats.Total.BytesSent

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1000
Configuration by String	IO.Stats.Total.BytesSent

This Tag returns the number of bytes sent since a Driver was loaded.

IO.Stats.Total.ConnectionCount

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1004
Configuration by String	IO.Stats.Total.ConnectionCount

This Tag returns the number of connections a Driver already established, successfully, since it was loaded.

IO.Stats.Total.TimeConnectedSeconds

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1002
Configuration by String	IO.Stats.Total.TimeConnectedSeconds

This Tag returns the number of seconds a Driver remained connected since it was loaded.

IO.Stats.Total.TimeDisconnectedSeconds

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	0 (zero)
N4 Parameter	1003
Configuration by String	IO.Stats.Total.TimeDisconnectedSeconds

This Tag returns the number of seconds a Driver remained disconnected since it was loaded.

Properties

Currently, there are no properties defined specifically to display I/O Interface statistics at run time.

Ethernet Interface Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of an **Ethernet** Interface.

I/O Tags

Tags of an Ethernet Interface (N2/B2 = 4)

The Tags described next allow controlling and identifying an **Ethernet** Interface at run time and they are also valid when the **RAS** Interface is selected.

IMPORTANT

These Tags are available **ONLY** while a Driver is in **Online** mode.

IO.Ethernet.IPSelect

Type of Tag	I/O Tag
Type of Access	Reading or Writing
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	4 (four)
N4 Parameter	0 (zero)
String Configuration	IO.Ethernet.IPSelect

Indicates the active IP address. Possible values are **0**: The main IP address is selected, **1**: The first alternative or backup IP address is selected, **2**: The second alternative or backup IP address is selected, or **3**: The third alternative or backup IP address is selected.

If the **Ethernet** or **RAS** Interface is connected, this Tag indicates which one of the four configured IP addresses is in use. If the Interface is disconnected, this Tag indicates which IP address is used first on the next attempt to connect.

During the connection process, if the active IP address is not available, the I/O Interface tries to connect using the other IP address. If the connection with the alternative IP address works, it is configured as the active IP address (automatic switchover).

To force a manual switchover, write values from 0 (zero) to 3 (three) to this Tag. This forces a reconnection with the specified IP address (**0**: Main address or **1, 2, 3**: Alternative address) if a Driver is currently connected. If a Driver is disconnected, this Tag configures the active IP address for the next attempt to connect.

IO.Ethernet.IPSwitch

Type of Tag	I/O Tag
Type of Access	Write-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	4 (four)
N4 Parameter	1 (one)
String Configuration	IO.Ethernet.IPSwitch

Any value written to this Tag forces a manual switchover. If the main IP address is active, then the first alternative or backup IP address is activated, and so on for all alternative IP addresses and returning to the main address until a connection is established.

If a Driver is disconnected, this Tag configures the active IP address for the next attempt to connect.

IO.Ethernet.SocketState

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	4 (four)
N4 Parameter	2 (two)
String Configuration	IO.Ethernet.SocketState

The Value property of this Tag corresponds to socket states as a map of bits:

- **Bit 0**: 0 (zero, not listening) or 1 (one, listening)
- **Bit 1**: 0 (zero, disconnected) or 1 (one, connected)

Properties

These properties control the configuration of an **Ethernet** Interface.

NOTE

The **Ethernet** Interface is also used by the **RAS** Interface.

IO.Ethernet.AcceptConnection

■ Configure to False if a Driver must not accept external connections, that is, if a Driver behaves as a master, or configure to True to enable the reception of connections, that is, if a Driver behaves as a slave.

IO.Ethernet.BackupEnable[2,3]

■ Configure to True to enable an alternative or backup IP address. If the reconnection attempt with the main IP address fails, a Driver tries to use an alternative IP address. Configure to False to disable its usage.

IO.Ethernet.BackupIP[2,3]

A Alternative or backup IP address of a remote device. Users can use a numerical address, as well as a device's host name, such as "192.168.0.7" or "SERVER2".

IO.Ethernet.BackupLocalPort[2,3]

9 Local port number to be used when connecting to an alternative IP address of a remote device. Used only if **IO.Ethernet.BackupLocalPortEnable** is equal to True.

IO.Ethernet.BackupLocalPortEnable[2,3]

■ Configure to True to force the use of a specific local port when connecting to an alternative or backup IP address or configure to False to use any available local port.

IO.Ethernet.BackupPort[2,3]

9 Port number of an alternative or backup IP address of a remote device, used with the **IO.Ethernet.BackupIP** property.

IO.Ethernet.IPFilter

A List with a comma-separated IPv4 or IPv6 addresses, which defines from which addresses a Driver accepts or blocks connections. Users can use asterisks, such as "192.168.*.*", or intervals, such as "192.168.0.41-50", in any part of IP addresses. To block an IP address or a range of IP addresses, use the tilde ("~") character at the beginning of the address, according to the next examples:

- **192.168.0.24**: Accepts only connections from IPv4 address 192.168.0.24
- **192.168.0.41-50**: Accepts connections from IPv4 addresses in the interval between 192.168.0.41 and 192.168.0.50
- **192.168.0.***: Accepts connections from IPv4 addresses in the interval between 192.168.0.0 and 192.168.0.255
- **fe80:3bf:877:::* (expands to fe80:03bf:0877:0000:0000:0000:0000:0000:*)**: Accepts connections from IPv6 addresses in the interval between fe80:03bf:0877:0000:0000:0000:0000:0000 and fe80:03bf:0877:0000:0000:0000:0000:ffff:ffff
- **192.168.0.10, 192.168.0.15, 192.168.0.20**: Accepts connections from IPv4 addresses 192.168.0.10, 192.168.0.15, and 192.168.0.20
- **~192.168.0.95, 192.168.0.***: Accepts connections from IPv4 addresses in the interval between 192.168.0.0 and 192.168.0.255, except the IPv4 address 192.168.0.95

When a Driver receives a connection attempt, the list of filters is scanned sequentially from left to right, searching for a specific authorization or block for the IP address where the connection comes from. If no element on the list corresponds to the IP address, the authorization or block are dictated by the last element of that list:

- If the last element on the list is an authorization, such as "192.168.0.24", then all IP addresses not found on the list are blocked
- If the last element on the list is a block, such as "~192.168.0.24", then all IP addresses not found on the list are authorized

If an IP address appears on more than one filter on the list, the leftmost filter has precedence. For example, in case of "~192.168.0.95, 192.168.0.*", the IP address 192.168.0.95 fits both rules, but the rule that wins is the leftmost one, "~192.168.0.95", and therefore this IP address is blocked.

When **IOKit** blocks a connection, it logs a message "Blocked incoming socket connection from {IP}!".

In case of UDP connections in broadcast listening mode, in which a Driver can receive packets from different IP addresses, blocks or permissions are performed at each packet received. If a packet is received from a blocked IP address, it logs a message "Blocked incoming packet from {IP} (discarding {N} bytes)!".

IO.Ethernet.ListenIP

 IP address of the local network interface that a Driver uses to establish and accept connections. Leave this property empty to establish and accepts connections using any local network interface.

IO.Ethernet.ListenPort

 Number of the IP port used by a Driver to listen to connections.

IO.Ethernet.MainIP

 IP address of a remote device. Users can use a numerical address, as well as a device's host name, such as "192.168.0.7" or "SERVER2".

IO.Ethernet.MainLocalPort

 Local port number to use when connecting to the main IP address of a remote device. This value is only used if the **IO.Ethernet.MainLocalPortEnable** property is equal to True.

IO.Ethernet.MainLocalPortEnable

 Configure to True to force the use of a specific local port when connecting to the main IP address of a remote device or configure to False to use any available local port.

IO.Ethernet.MainPort

 Number of the IP port of a remote device, used with the **IO.Ethernet.MainIP** property.

IO.Ethernet.PingEnable

 Configure to True to enable sending a **ping** command to the IP address of a remote device, before trying to connect to the socket. This socket's connection time-out cannot be controlled, therefore sending a **ping** command before connecting is a fast way to detect if the connection is going to fail. Configure to False to disable a **ping** command.

IO.Ethernet.PingTimeoutMs

 Delay time to wait for a response from a **ping** command, in milliseconds.

IO.Ethernet.PingTries

 Maximum number of attempts of a **ping** command. Minimum value is 1 (one), including the first **ping** command.

IO.Ethernet.ShareListenPort

 Configure to True to share a listening port with other Drivers and processes or False to open a listening port in exclusive mode. To successfully share a listening port, all Drivers and processes that use that port must open it in shared mode. When a listening port is shared, each incoming connection is distributed to one of the processes listening. This way, if a Slave Driver only supports one connection at a time, users can use several instances of this Driver listening on the same port, therefore simulating a Driver with support for multiple connections.

IO.Ethernet.SuppressEcho

 Configure to True to eliminate echoes in communication. An echo is the unwanted reception of an exact copy of all data packets a Driver sent to a device.

IO.Ethernet.Transport

 Defines a transport protocol. Possible values are **T** or **TCP**: Uses the TCP/IP protocol or **U** or **UDP**: Uses the UDP/IP protocol.

IO.Ethernet.UseIPv6

 Configure to True to use IPv6 addresses on all Ethernet connections or configure to False to use IPv4 addresses (default).

Modem Interface Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of a **Modem** (TAPI) Interface.

I/O Tags

Tags of a Modem Interface (N2/B2 = 3)

The Tags described next allow controlling and diagnosing a **Modem** (TAPI) Interface at run time.

IMPORTANT

These Tags are available **ONLY** while a Driver is in **Online** mode.

IO.TAPI.ConnectionBaudRate

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	5 (five)
String Configuration	IO.TAPI.ConnectionBaudRate

Indicates a baud rate value for the current connection. If a modem is not connected, returns the value 0 (zero).

IO.TAPI.Dial

Type of Tag	I/O Tag
Type of Access	Write-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	1 (one)
String Configuration	IO.TAPI.Dial

Write any value to this Tag to force a **Modem** Interface to start a call. This is an asynchronous command, only starting the call process. Users can monitor the **IO.TAPI.IsModemConnected** Tag to detect when a call is established.

IO.TAPI.HangUp

Type of Tag	I/O Tag
Type of Access	Write-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	4 (four)
String Configuration	IO.TAPI.HangUp

Any value written to this Tag hangs the current call up.

NOTE

Use this command only when managing the physical layer manually or when explicitly trying to force a Driver to restart the communication. If the physical layer is configured for automatic reconnection, a Driver immediately tries to reestablish the connection.

IO.TAPI.IsModemConnected

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	3 (three)
String Configuration	IO.TAPI.IsModemConnected

This Tag indicates the status of a modem connection. Possible values are **0**: The modem is not connected, but it may be performing or receiving an external call or **1**: The modem is connected and a Driver completed or received an external call successfully. While it is in this status, the physical layer can send or receive data.

IO.TAPI.IsModemConnecting

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	6 (six)
String Configuration	IO.TAPI.IsModemConnecting

This Tag indicates the status of a modem connection, with more details than the **IO.TAPI.IsModemConnected** Tag. Possible values are **0**: Modem is not connected, **1**: Modem is connecting, that is, performing or receiving an external call, **2**: Modem is connected. While in this status, the physical layer can send or receive data, or **3**: Modem is disconnecting the current call.

IO.TAPI.ModemStatus

Type of Tag	I/O Tag
Type of Access	Read-Only
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	2 (two)
String Configuration	IO.TAPI.ModemStatus

Returns a **String** with the current status of a modem. Possible values are the following:

- **"No status!"**: The **Modem** Interface was not open yet or was already closed
- **"Modem initialized OK!"**: The **Modem** Interface was initialized successfully
- **"Modem error at initialization!"**: A Driver could not initialize modem's line. Check that Driver's log file for more details
- **"Modem error at dial!"**: A Driver could not start or accept a call
- **"Connecting..."**: A Driver started a call successfully, and is currently processing that call
- **"Ringing..."**: Indicates that the modem is receiving an external call, but it did not accepted it yet
- **"Connected!"**: A Driver connected successfully, that is, completed or accepted an external call
- **"Disconnecting..."**: A Driver is turning the current call off
- **"Disconnected OK!"**: A Driver turned the current call off
- **"Error: no dial tone!"**: A Driver aborted a call because the available line signal was not detected
- **"Error: busy!"**: A Driver aborted a call because the line was busy
- **"Error: no answer!"**: A Driver aborted a call because no answer was received from the other modem
- **"Error: unknown!"**: Current call was aborted because of an unknown error

IO.TAPI.PhoneNumber

Type of Tag	I/O Tag
Type of Access	Reading or Writing
N1 Parameter	-1 (minus one)
N2 Parameter	0 (zero)
N3 Parameter	3 (three)
N4 Parameter	0 (zero)
String Configuration	IO.TAPI.PhoneNumber

This Tag is a **String** that reads or changes the telephone number used by the **IO.TAPI.Dial** Tag. When changing this Tag, the new value is used only on the next **Dial** command.

Properties

These properties control the configuration of a **Modem** (TAPI) Interface.

IO.TAPI.AcceptIncoming

9 Configure to False if a modem cannot accept external calls, that is, if a Driver behaves as a master, and configure to True to enable receiving calls, that is, if a Driver behaves as a slave.

IO.TAPI.ModemID

9 This is the identification number of a modem. This ID is created by Windows and used internally to identify a modem on a list of devices installed on a computer. This ID may not remain valid if a modem is reinstalled or an application is executed on another computer.

NOTE

It is advisable to configure this property as 0 (zero), indicating that a Driver must use the first available modem.

IO.TAPI.PhoneNumber

A A telephone number used by **Dial** commands, such as "0w01234566", in which the "w" character forces a modem to wait for a call sign.

RAS Interface Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of a **RAS** Interface.

I/O Tags

Tags of a RAS Interface (N2/B2 = 5)

Currently, there are no Tags defined specifically to manage a **RAS** Interface at run time.

Properties

These properties control the configuration of a **RAS** Interface.

NOTE

A **RAS** Interface uses the **Ethernet** Interface, which therefore must be also configured.

IO.RAS.ATCommand

A An **AT** command to send through a socket to force a RAS device to perform a call using the current RAS channel, such as "ATDT6265545".

IO.RAS.CommandTimeoutSec

9 Time to wait for a **CONNECT** message in response to an **AT** command, in seconds.

Serial Interface Configuration

This section contains information about the configuration of **I/O Tags** and **Properties** of a **Serial** Interface.

I/O Tags

Tags of a Serial Interface (N2/B2 = 2)

Currently, there are no Tags defined specifically to manage a **Serial** Interface at run time.

Properties

These properties control the configuration of a **Serial** Interface.

IO.Serial.Baudrate

9 Specifies a baud rate of a serial port, such as 9600.

IO.Serial.CTSTimeoutMs

9 Time to wait for a **CTS** signal, in milliseconds. After turning the **RTS** signal on, a timer is started to wait for a **CTS** signal. If this timer expires, a Driver aborts sending bytes through the serial port. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle** and the **IO.Serial.WaitCTS** property is configured to True.

IO.Serial.DataBits

9 Specifies the number of data bits to configure a serial port. Possible values are **5**: Five data bits, **6**: Six data bits, **7**: Seven data bits, or **8**: Eight data bits.

IO.Serial.DelayAfterMs

9 Number of milliseconds to delay after the last byte is sent through a serial port, but before turning the **RTS** signal off. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle** and the **IO.Serial.WaitCTS** property is configured to False.

IO.Serial.DelayBeforeMs

9 Number of milliseconds to delay after turning the **RTS** signal on, but before data is sent. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle** and the **IO.Serial.WaitCTS** property is configured to False.

IO.Serial.DTR

A Indicates how a Driver deals with the **DTR** signal. Possible values are **OFF**: **DTR** signal is always turned off or **ON**: **DTR** signal is always turned on.

IO.Serial.InterbyteDelayUs

9 Delay time, in milliseconds (1/1000000 of a second), for each two bytes sent through a **Serial** Interface.

IO.Serial.InterframeDelayMs

9 Delay time, in milliseconds, before sending a packet after the last packet sent or received.

IO.Serial.Parity

A Specifies a parity for the configuration of a serial port. Possible values are **E or Even**: Even parity, **N or None**: No parity, **O or Odd**: Odd parity, **M or Mark**: Mark parity, or **S or Space**: Space parity.

IO.Serial.Port

9 Number of the local serial port. Possible values are **1**: Uses the COM1 port, **2**: Uses the COM2 port, **3**: Uses the COM3 port, or **n**: Uses the COMn port.

IO.Serial.RTS

A Indicates how a Driver deals with the **RTS** signal. Possible values are **OFF**: **RTS** signal always off, **ON**: **RTS** signal always on, or **Toggle**: Turns the **RTS** signal on when transmitting data and turns the **RTS** signal off when not transmitting data.

IO.Serial.StopBits

9 Specifies the number of stop bits for the configuration of a serial port. Possible values are **1**: One stop bit, **2**: One and a half stop bit, or **3**: Two stop bits.

IO.Serial.SuppressEcho

9 Use a value different from 0 (zero) to enable suppressing the echo or 0 (zero) to disable it.

IO.Serial.WaitCTS

■ Configure to True to force a Driver to wait for the **CTS** signal before sending bytes when the **RTS** signal is turned on. Available only when the **IO.Serial.RTS** property is configured with the value **Toggle**.

Appendix I - BACnet Protocol

BACnet objects supported by the driver

The following BACnet objects are supported by the driver:

- **accessCredential**
- **accessDoor**
- **accessPoint**
- **accessRights**
- **accessUser**
- **accessZone**
- **accumulator**
- **alertEnrollment**
- **analogInput**

- **analogOutput**
- **analogValue**
- **averaging**
- **binaryInput**
- **binaryLightingOutput**
- **binaryOutput**
- **binaryValue**
- **bitstringValue**
- **calendar**
- **channel**
- **characterstringValue**
- **command**
- **credentialDataInput**
- **dateValue**
- **datepatternValue**
- **datetimeValue**
- **datetimepatternValue**
- **device**
- **elevatorGroup**
- **escalator**
- **eventEnrollment**
- **eventLog**
- **file**
- **globalGroup**
- **group**
- **integerValue**
- **largeAnalogValue**
- **lifeSafetyPoint**
- **lifeSafetyZone**
- **lift**
- **lightingOutput**
- **loadControl**

- **loop**
- **multiStateInput**
- **multiStateOutput**
- **multiStateValue**
- **networkPort**
- **networkSecurity**
- **notificationClass**
- **notificationForwarder**
- **octetstringValue**
- **positiveIntegerValue**
- **program**
- **pulseConverter**
- **schedule**
- **structuredView**
- **timeValue**
- **timepatternValue**
- **timer**
- **trendLog**
- **trendLogMultiple**

accessCredential Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a credential that is used for authentication and authorization when requesting access. The credential can be owned by an access user of any type. Access user ownership is represented by a reference to an Access User object.

Objects of type accessCredential have the following properties:

- **absenteeLimit**
- **activationTime**
- **assignedAccessRights**
- **authenticationFactors**
- **authorizationExemptions**
- **belongsTo**
- **credentialDisable**

- **credentialStatus**
- **daysRemaining**
- **description**
- **expirationTime**
- **extendedTimeEnable**
- **globalIdentifier**
- **lastAccessEvent**
- **lastAccessPoint**
- **lastUseTime**
- **objectIdentifier**
- **objectName**
- **objectType**
- **profileLocation**
- **profileName**
- **propertyList**
- **reasonForDisable**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**
- **tags**
- **threatAuthority**
- **traceFlag**
- **usesRemaining**

accessCredential.absenteeLimit

According to the BACnet protocol documentation:

Specifies the maximum number of consecutive days for which the credential can remain inactive (i.e. unused) before it becomes disabled.

Property *absenteeLimit* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.absenteeLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 244

Tag Value

A 32-bit unsigned integer.

accessCredential.activationTime

According to the BACnet protocol documentation:

Specifies the date and time at or after which the credential becomes active. If the current time is before the activation time, the credential shall be disabled and the value DISABLED_NOT_YET_ACTIVE shall be added to the Reason_For_Disable list. The value DISABLED_NOT_YET_ACTIVE shall be removed from the list when this condition no longer applies. If all of the octets of the BACnetDateTime value contain a value of X'FF', then the credential has an activation time of 'start of time'.

Property *activationTime* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.activationTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 254

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

accessCredential.assignedAccessRights

According to the BACnet protocol documentation:

Specifies the access rights assigned to this credential.

Property *assignedAccessRights* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.assignedAccessRights

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 256

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAssignedAccessRights>**.

accessCredential.authenticationFactors

According to the BACnet protocol documentation:

Specifies the authentication factors that belong to this credential. Any access attempt using an authentication factor which is disabled shall fail. In this case, the Access_Event property of the Access Point object where this authentication factor was used shall be set to the value corresponding to the reason why it was disabled.

Property *authenticationFactors* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.authenticationFactors

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 257

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetCredentialAuthenticationFactor>**.

accessCredential.authorizationExemptions

According to the BACnet protocol documentation:

The authorization checks from which this credential is exempt. When a credential is exempt from an authorization check, the access attempt shall not be denied due to this authorization criterion.

Property *authorizationExemptions* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.authorizationExemptions

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 364

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAuthorizationExemption>**.

accessCredential.belongsTo

According to the BACnet protocol documentation:

References an Access User object that represents the owning access user (i.e. person, group, or asset).

Property *belongsTo* of object *accessCredential* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.belongsTo
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 32
B3:	<objectAddress>
B4:	<property> = 262
Size:	2 elements

Block Elements

Index	Name	Type
0	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.
1	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessCredential.credentialDisable

According to the BACnet protocol documentation:

Contains a value that disables a credential for reasons external to this object. If this property is writable, then it is the mechanism by which an operator or external process may disable the credential. When this property is changed, any disable reason added to the Reason_For_Disable list as a result of a previous change of this property shall be removed from that list. When this property takes on any value other than NONE, the corresponding disable-reason value shall be added to the Reason_For_Disable list.

Property *credentialDisable* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.credentialDisable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 263

Tag Value

An integer representing one of the options of enumeration **<BACnetAccessCredentialDisable>**.

accessCredential.credentialStatus

According to the BACnet protocol documentation:

Specifies whether the credential is active or inactive. Only the value ACTIVE enables the credential to be used for authentication. While the list in property Reason_For_Disable is nonempty, the status of the credential shall be INACTIVE, otherwise it shall be ACTIVE.

Property *credentialStatus* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.credentialStatus

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 264

Tag Value

An integer representing one of the options of enumeration **<BACnetBinaryPV>**.

accessCredential.daysRemaining

According to the BACnet protocol documentation:

Specifies the number of remaining days for which the credential can be used. If this property has a value greater than zero, its value shall be decremented by one when the credential this object represents is granted access at an access controlled point, and the current date is more recent than the date indicated in the property Last_Use_Time. If this property becomes zero, the Access Credential shall be disabled and the value DISABLED_MAX_DAYS shall be added to the Reason_For_Disable property. The value DISABLED_MAX_DAYS shall be removed from the Reason_For_Disable property when this property is set to a value greater than zero. If this property is present and the credential this object represents is not limited in the days it can be used, then the value of this property shall be -1 and DISABLED_MAX_USES shall never be added to the Reason_For_Disable property.

Property *daysRemaining* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.daysRemaining

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 267

Tag Value

A 32-bit signed integer.

accessCredential.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

accessCredential.expirationTime

According to the BACnet protocol documentation:

Specifies the date and time after which the credential will expire. This defines the end of the validity period of the credential. If the current time is after the expiry time, the credential shall be disabled and the value `DISABLED_EXPIRED` shall be added to the `Reason_For_Disable` list. The value `DISABLED_EXPIRED` shall be removed from the list when this condition no longer applies. If all of the fields of the `BACnetDateTime` value contain a value of `X'FF'`, then the credential has an expiry time of 'end-of-time'.

Property `expirationTime` of object `accessCredential` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.expirationTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 270

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;

- A character string containing an XML with a value of type **<BACnetDateTime>**.

accessCredential.extendedTimeEnable

According to the BACnet protocol documentation:

Specifies which command of type BACnetDoorValue shall be used to command the access door when access is granted. If extended time is enabled (TRUE), EXTENDED_PULSE_UNLOCK is used, otherwise (FALSE) PULSE_UNLOCK is used.

Property *extendedTimeEnable* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.extendedTimeEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 271

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessCredential.globalIdentifier

According to the BACnet protocol documentation:

A unique identifier which is used to globally identify the access controlled zone this object represents. This value may be used to identify Access Zone objects in multiple devices that represent the same access controlled zone. If this value is assigned, it shall be unique internetwork-wide and all Access Zone objects in all devices that represent this access controlled zone shall have this value. A value of zero indicates that no global identifier is assigned.

Property *globalIdentifier* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.globalIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 323

Tag Value

A 32-bit unsigned integer.

accessCredential.lastAccessEvent

According to the BACnet protocol documentation:

Specifies the last access event generated at an access controlled point upon use of this credential. If the credential this object represents has never been used, then this property shall have a value of NONE.

Property *lastAccessEvent* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.lastAccessEvent

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 275

Tag Value

An integer representing one of the options of enumeration **<BACnetAccessEvent>**.

accessCredential.lastAccessPoint

According to the BACnet protocol documentation:

Refers to the last Access Point object where one of the authentication factors of the credential has been used. If property level COV is in effect for this property, any update of this property shall cause a COV notification to be issued, regardless of whether the value of this property changes. If the credential this object represents has never been used, then this property shall contain 4194303 in the instance part of the object identifier and in the device instance part of the device identifier, if present.

Property *lastAccessPoint* of object *accessCredential* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.lastAccessPoint
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 32
B3:	<objectAddress>
B4:	<property> = 276
Size:	2 elements

Block Elements

Index	Name	Type
0	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.
1	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessCredential.lastUseTime

According to the BACnet protocol documentation:

Specifies the date and time of the last use of the credential at an access controlled point, independent of whether access was granted or denied. If the credential this object represents has never been used, then this property shall have the value X'FF' for all date and time octets.

Property *lastUseTime* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.lastUseTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 281

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

accessCredential.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessCredential.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

accessCredential.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

accessCredential.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

accessCredential.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

accessCredential.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

accessCredential.reasonForDisable

According to the BACnet protocol documentation:

Contains a list of disable-reasons why the credential has been disabled. The credential can be disabled for multiple reasons at the same time. While the Credential_Status property has a value INACTIVE, this list shall not be empty. When an entry is removed from this list that results in the list becoming empty, the Credential_Status shall be set to ACTIVE.

Property *reasonForDisable* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.reasonForDisable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 303

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAccessCredentialDisableReason>**.

accessCredential.reliability

According to the BACnet protocol documentation:

Indicates whether the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

accessCredential.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessCredential.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *accessCredential* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 32
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessCredential.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

accessCredential.threatAuthority

According to the BACnet protocol documentation:

Specifies the maximum threat level for which this credential is valid. If this value is less than the Threat_Level property of the Access Point object where the access credential is used, access is denied. If this property is not present, the threat authority of this credential is assumed to be zero.

Property *threatAuthority* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.threatAuthority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 306

Tag Value

A 32-bit unsigned integer.

accessCredential.traceFlag

According to the BACnet protocol documentation:

Specifies whether the credential is being traced. When a traced credential is used at an access point, the Access_Event property of the corresponding Access Point object shall be set to TRACE.

Property *traceFlag* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.traceFlag

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 308

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessCredential.usesRemaining

According to the BACnet protocol documentation:

Specifies the number of remaining uses that the credential can be used for authentication. If this property has a value greater than zero and access is granted at an access controlled point, then the value of this property shall be decremented by one. If this property becomes zero, then the Access Credential shall be disabled and the value DISABLED_MAX_USES shall be added to the Reason_For_Disable property. The value DISABLED_MAX_USES shall be removed from the Reason_For_Disable property when this property is set to a value greater than zero. If this property is present and the credential this object represents is not limited in the number of uses, then the value of this property shall be -1 and DISABLED_MAX_USES shall never be added to the Reason_For_Disable property.

Property *usesRemaining* of object *accessCredential* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessCredential-<objectAddress>.usesRemaining

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 32
N3:	<objectAddress>
N4:	<property> = 319

Tag Value

A 32-bit signed integer.

accessDoor Object

According to the BACnet protocol documentation:

Abstract interface to a physical door whose properties represent the externally visible characteristics of an access control door. The Access Door is comprised of a collection of physical door hardware, such as a door lock, a door contact, and a Request-To-Exit device, which together comprise a door for access control. The individual hardware components of the door may or may not be exposed through this object.

Objects of type `accessDoor` have the following properties:

- **ackedTransitions**
- **alarmValues**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **doorAlarmState**
- **doorExtendedPulseTime**
- **doorMembers**
- **doorOpenTooLongTime**
- **doorPulseTime**
- **doorStatus**
- **doorUnlockDelayTime**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultValues**
- **lastCommandTime**
- **lockStatus**
- **maintenanceRequired**
- **maskedAlarmValues**
- **notificationClass**
- **notifyType**
- **objectIdentifier**

- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **securedStatus**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **valueSource**
- **valueSourceArray**

accessDoor.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *accessDoor* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 30
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessDoor.alarmValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-OFFNORMAL event is generated.

Property *alarmValues* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.alarmValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 7

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDoorAlarmState>**.

accessDoor.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *accessDoor* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 30
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

accessDoor.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

accessDoor.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

accessDoor.doorAlarmState

According to the BACnet protocol documentation:

Indicates the alarm state for the physical door and is restricted to the values NORMAL and those contained in Alarm_Values and Fault_Values. When no alarm or fault condition exists for this object, this property shall take on the value NORMAL. It is considered a local matter as to when this property is set to a non-normal value. It is up to the internal control logic to take Lock_Status, Door_Status, Present_Value and information from other objects into account when calculating the proper alarm state. However, this property cannot take on any value which is also in the Masked_Alarm_Values list. If the property is currently set to a specific state and that state is written to the Masked_Alarm_Values list, then the Door_Alarm_State will immediately return to the NORMAL state. This property, if present, is required to be writable when Out_Of_Service is TRUE.

Property *doorAlarmState* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.doorAlarmState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 226

Tag Value

An integer representing one of the options of enumeration **<BACnetDoorAlarmState>**.

accessDoor.doorExtendedPulseTime

According to the BACnet protocol documentation:

Indicates the maximum amount of time, in tenths of seconds, which the door will be unlocked when the Present_Value has a value of EXTENDED_PULSE_UNLOCK, after which time the Present_Value shall be automatically relinquished at the priority that established the EXTENDED_PULSE_UNLOCK command.

Property *doorExtendedPulseTime* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.doorExtendedPulseTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 227

Tag Value

A 32-bit unsigned integer.

accessDoor.doorMembers

According to the BACnet protocol documentation:

Holds an array of references to BACnet objects which represent I/O devices, authentication devices, schedules, programs, or other objects that are associated with the physical door. It is a local matter as to how this array is used and which objects are referenced in this array. The array may be empty or not present if the vendor does not wish to expose the individual objects that make up this physical door.

Property *doorMembers* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.doorMembers

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 228

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectReference>**.

accessDoor.doorOpenTooLongTime

According to the BACnet protocol documentation:

Indicates the time, in tenths of seconds, to delay before setting the Door_Alarm_State to DOOR_OPEN_TOO_LONG after it is determined that a door-open-too-long condition exists.

Property *doorOpenTooLongTime* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.doorOpenTooLongTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 229

Tag Value

A 32-bit unsigned integer.

accessDoor.doorPulseTime

According to the BACnet protocol documentation:

Indicates the maximum duration of time, in tenths of seconds, for which the door will be unlocked when the Present_Value has a value of PULSE_UNLOCK, after which time the Present_Value shall be automatically relinquished at the priority that established the PULSE_UNLOCK command.

Property *doorPulseTime* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.doorPulseTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 230

Tag Value

A 32-bit unsigned integer.

accessDoor.doorStatus

According to the BACnet protocol documentation:

Represents the open or closed state of the physical door.

Property *doorStatus* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.doorStatus

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 231

Tag Value

An integer representing one of the options of enumeration **<BACnetDoorStatus>**.

accessDoor.doorUnlockDelayTime

According to the BACnet protocol documentation:

Indicates the duration of time, in tenths of seconds, which the physical door lock will delay unlocking when the Present_Value changes to a value of PULSE_UNLOCK or EXTENDED_PULSE_UNLOCK.

Property *doorUnlockDelayTime* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.doorUnlockDelayTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 232

Tag Value

A 32-bit unsigned integer.

accessDoor.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `accessDoor` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessDoor.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property `eventAlgorithmInhibitRef` of object `accessDoor` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

accessDoor.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessDoor.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *accessDoor* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 30
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessDoor.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *accessDoor* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 30
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

accessDoor.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *accessDoor* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 30
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

accessDoor.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

accessDoor.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *accessDoor* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 30
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

accessDoor.faultValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-FAULT event is generated.

Property *faultValues* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.faultValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 39

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDoorAlarmState>**.

accessDoor.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

accessDoor.lockStatus

According to the BACnet protocol documentation:

Represents the monitored (as opposed to the commanded) status of the door lock.

Property *lockStatus* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.lockStatus

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 233

Tag Value

An integer representing one of the options of enumeration **<BACnetLockStatus>**.

accessDoor.maintenanceRequired

According to the BACnet protocol documentation:

Indicates that maintenance is required for one or more of the life safety points that are members of this zone.

Property *maintenanceRequired* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.maintenanceRequired

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 158

Tag Value

An integer representing one of the options of enumeration **<BACnetMaintenance>**.

accessDoor.maskedAlarmValues

According to the BACnet protocol documentation:

Specifies any alarm and/or fault states which are masked. An alarm state which is currently masked will prevent the Door_Alarm_State property from being equal to that state.

Property *maskedAlarmValues* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.maskedAlarmValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 234

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDoorAlarmState>**.

accessDoor.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

accessDoor.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

accessDoor.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessDoor.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

accessDoor.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

accessDoor.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessDoor.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

An integer representing one of the options of enumeration **<BACnetDoorValue>**.

accessDoor.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *accessDoor* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 30
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
1	elm01	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
2	elm02	One of the following data types:

accessDoor.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

accessDoor.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

accessDoor.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

accessDoor.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

accessDoor.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessDoor.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

An integer representing one of the options of enumeration **<BACnetDoorValue>**.

accessDoor.securedStatus

According to the BACnet protocol documentation:

Represents whether or not the physical door is in a secured state.

Property *securedStatus* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.securedStatus

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 235

Tag Value

An integer representing one of the options of enumeration **<BACnetDoorSecuredStatus>**.

accessDoor.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *accessDoor* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 30
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessDoor.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

accessDoor.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

accessDoor.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

accessDoor.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *accessDoor* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 30
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

accessDoor.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *accessDoor* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessDoor-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 30
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

accessPoint Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics associated with the authentication and authorization process of an access controlled point. (e.g., door, gate, turnstile). Access through this point is directional in that it represents access in one direction only. A door, in which access is controlled in both directions, is represented by two separate Access Point objects.

Objects of type accessPoint have the following properties:

- **accessAlarmEvents**
- **accessDoors**
- **accessEvent**
- **accessEventAuthenticationFactor**
- **accessEventCredential**
- **accessEventTag**
- **accessEventTime**
- **accessTransactionEvents**
- **accompanimentTime**
- **ackedTransitions**
- **activeAuthenticationPolicy**
- **authenticationPolicyList**
- **authenticationPolicyNames**
- **authenticationStatus**
- **authorizationMode**
- **description**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **failedAttemptEvents**
- **failedAttempts**

- **failedAttemptsTime**
- **lockout**
- **lockoutRelinquishTime**
- **maxFailedAttempts**
- **musterPoint**
- **notificationClass**
- **notifyType**
- **numberOfAuthenticationPolicies**
- **objectIdentifier**
- **objectName**
- **objectType**
- **occupancyCountAdjust**
- **occupancyLowerLimitEnforced**
- **occupancyUpperLimitEnforced**
- **outOfService**
- **priorityForWriting**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**
- **tags**
- **threatLevel**
- **transactionNotificationClass**
- **verificationTime**
- **zoneFrom**
- **zoneTo**

accessPoint.accessAlarmEvents

According to the BACnet protocol documentation:

The value of this property is used as the value of the pAccessEvents parameter of the object's ACCESS_EVENT event algorithm for Access Alarm Events.

Property *accessAlarmEvents* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.accessAlarmEvents

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 245

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAccessEvent>**.

accessPoint.accessDoors

According to the BACnet protocol documentation:

Specifies the references to those Access Door objects whose Present_Value properties are commanded after successful authorization. If this Access Point object does not command Access Door objects (e.g., muster point), or is used to control access to other resources or functions, or commands other objects, then this array shall be empty.

Property *accessDoors* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.accessDoors

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 246

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectReference>**.

accessPoint.accessEvent

According to the BACnet protocol documentation:

Indicates the last access event which occurred at this Access Point.

Property *accessEvent* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.accessEvent

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 247

Tag Value

An integer representing one of the options of enumeration **<BACnetAccessEvent>**.

accessPoint.accessEventAuthenticationFactor

According to the BACnet protocol documentation:

Specifies the authentication factor that corresponds to the access event specified in the Access_Event property, if applicable. Otherwise it shall contain a value of format type UNDEFINED.

Property *accessEventAuthenticationFactor* of object *accessPoint* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.accessEventAuthenticationFactor
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 33
B3:	<objectAddress>
B4:	<property> = 248
Size:	3 elements

Block Elements

Index	Name	Type
0	FormatType	An integer representing one of the options of enumeration <BACnetAuthenticationFactorType> .
1	FormatClass	A 32-bit unsigned integer.
2	Value	A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

accessPoint.accessEventCredential

According to the BACnet protocol documentation:

Specifies the Access Credential object that corresponds to the access event specified in the Access_Event property, if applicable. This property shall contain 4194303 in the instance part of the object identifier and in the device instance part of the device identifier, if present, under the following conditions: there is no credential recognized up to now, or there is no credential that is associated to the current access event, or the credential of the authentication factor that is associated to the current event is unknown, or the device chooses not to expose the credential.

Property *accessEventCredential* of object *accessPoint* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.accessEventCredential
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 33
B3:	<objectAddress>
B4:	<property> = 249
Size:	2 elements

Block Elements

Index	Name	Type
0	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.
1	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessPoint.accessEventTag

According to the BACnet protocol documentation:

Numeric value which identifies the access transaction to which the current access event belongs. Multiple access events may be generated by a single access transaction. The value of this property shall increase monotonically for each new access transaction. It may be implemented using modulo arithmetic.

Property *accessEventTag* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.accessEventTag

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 322

Tag Value

A 32-bit unsigned integer.

accessPoint.accessEventTime

According to the BACnet protocol documentation:

Indicates the most recent update time of the Access_Event property. This property shall update its value on each update of Access_Event. Update times of type Time or Date shall have X'FF' in each octet, and Sequence Number update times shall have the value 0 if no update has yet occurred.

Property *accessEventTime* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.accessEventTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 250

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

accessPoint.accessTransactionEvents

According to the BACnet protocol documentation:

Used as the value of the pAccessEvents parameter of the object's ACCESS_EVENT event algorithm for Access Transaction Events.

Property *accessTransactionEvents* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.accessTransactionEvents

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 251

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAccessEvent>**.

accessPoint.accompanimentTime

According to the BACnet protocol documentation:

Specify the time, in seconds, to wait for a second credential to be presented at this access point when the original credential requires accompaniment. If an accompanying credential is not presented within this time the authorization of the original credential shall fail and the Access_Event property shall be set to DENIED_NO_ACCOMPANIMENT.

Property *accompanimentTime* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.accompanimentTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 253

Tag Value

A 32-bit unsigned integer.

accessPoint.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *accessPoint* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 33
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessPoint.activeAuthenticationPolicy

According to the BACnet protocol documentation:

Specifies the active authentication policy. The active authentication policy of this object shall be one of 'n' authentication policies, where 'n' is the number of authentication policies defined in the Number_Of_Authentication_Policies property.

Property *activeAuthenticationPolicy* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.activeAuthenticationPolicy

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 255

Tag Value

A 32-bit unsigned integer.

accessPoint.authenticationPolicyList

According to the BACnet protocol documentation:

Specifies the authentication policies defined for this Access Point.

Property *authenticationPolicyList* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.authenticationPolicyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 258

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAuthenticationPolicy>**.

accessPoint.authenticationPolicyNames

According to the BACnet protocol documentation:

Specifies the names of the defined authentication policies.

Property *authenticationPolicyNames* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.authenticationPolicyNames

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 259

Tag Value

A character string containing an XML with a value of type **<SequenceOfCharacterString>**.

accessPoint.authenticationStatus

According to the BACnet protocol documentation:

indicates the current status of the authentication process.

Property *authenticationStatus* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.authenticationStatus

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 260

Tag Value

An integer representing one of the options of enumeration **<BACnetAuthenticationStatus>**.

accessPoint.authorizationMode

According to the BACnet protocol documentation:

Determines how authorization is performed at the Access Point. An Access Point object is not required to support all of these authorization modes but is required to support at least AUTHORIZE.

Property *authorizationMode* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.authorizationMode

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 261

Tag Value

An integer representing one of the options of enumeration **<BACnetAuthorizationMode>**.

accessPoint.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

accessPoint.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `accessPoint` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessPoint.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property `eventAlgorithmInhibitRef` of object `accessPoint` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

accessPoint.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessPoint.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *accessPoint* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 33
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessPoint.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *accessPoint* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 33
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

accessPoint.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *accessPoint* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 33
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

accessPoint.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

accessPoint.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *accessPoint* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 33
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

accessPoint.failedAttemptEvents

According to the BACnet protocol documentation:

Specifies those access events that are counted as a failed access attempt.

Property *failedAttemptEvents* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.failedAttemptEvents

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 272

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAccessEvent>**.

accessPoint.failedAttempts

According to the BACnet protocol documentation:

Indicates the current count of successive failed access attempts. Any successive failed access attempt shall increment the value of this property. This property shall be set to zero when a successful access attempt occurs or when the property Lockout becomes FALSE.

Property *failedAttempts* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.failedAttempts

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 273

Tag Value

A 32-bit unsigned integer.

accessPoint.failedAttemptsTime

According to the BACnet protocol documentation:

Specifies the time, in seconds, to delay before setting the Failed_Attempts property to zero, after the last failed access attempt.

Property *failedAttemptsTime* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.failedAttemptsTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 274

Tag Value

A 32-bit unsigned integer.

accessPoint.lockout

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the access controlled point this object represents is in a lockout state. When the access point is in a lockout state, any access request shall always be denied, except for an active credential for which the value LOCKOUT is contained in the Authorization_Exemptions property of the corresponding Access Credential object. For each denied access request, the Access_Event property shall be set to DENIED_LOCKOUT. An Access Point object may be set to a lockout state due to too many failed access attempts, as defined in the Max_Failed_Attempts property, or by writing TRUE to this property. When the property Lockout becomes TRUE due to too many failed access attempts, then the Access_Event property shall be set to LOCKOUT_MAX_ATTEMPTS. If TRUE is written to this property for any other reason, the Access_Event property shall be set to LOCKOUT_OTHER. When the Lockout property becomes FALSE, the Access_Event property shall be set to LOCKOUT_RELINQUISHED.

Property *lockout* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.lockout

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 282

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessPoint.lockoutRelinquishTime

According to the BACnet protocol documentation:

Specifies the time, in seconds, to delay after the Lockout property has taken on the value TRUE, before automatically relinquishing the lockout state. The lockout state is relinquished by setting the Lockout property to FALSE. A value of zero indicates that the lockout state will not automatically be relinquished.

Property *lockoutRelinquishTime* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.lockoutRelinquishTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 283

Tag Value

A 32-bit unsigned integer.

accessPoint.maxFailedAttempts

According to the BACnet protocol documentation:

Specifies the maximum number of successive failed access attempts before the Lockout property is set to TRUE. If the Failed_Attempts property becomes greater than or equal to the value of this property and this property is not zero, the Lockout property is set to TRUE. Zero indicates that the Lockout property is not set to TRUE as the result of failed access attempts.

Property *maxFailedAttempts* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.maxFailedAttempts

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 285

Tag Value

A 32-bit unsigned integer.

accessPoint.musterPoint

According to the BACnet protocol documentation:

Indicates whether this Access Point generates muster access events (TRUE) or not (FALSE). A muster event is generated by setting the Access_Event property to MUSTER after an access credential has been presented at the access point. It is a local matter as to whether a muster event is generated for unknown credentials.

Property *musterPoint* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.musterPoint

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 287

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessPoint.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

accessPoint.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

accessPoint.numberOfAuthenticationPolicies

According to the BACnet protocol documentation:

Specifies the number of specified authentication policies. This property shall always have a value greater than zero. If the value of this property is changed, the size of the Authentication_Policy_List array and the size of the Authentication_Policy_Names array, if present, shall also be changed to the same value.

Property *numberOfAuthenticationPolicies* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.numberOfAuthenticationPolicies

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 289

Tag Value

A 32-bit unsigned integer.

accessPoint.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessPoint.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

accessPoint.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

accessPoint.occupancyCountAdjust

According to the BACnet protocol documentation:

Indicates whether (TRUE) this object will adjust the occupancy count of the zones for which it controls access, or not (FALSE). The occupancy count is decremented for the zone for which this Access Point is an exit point and incremented for the zone for which this Access Point is an entry point. Occupancy count shall be adjusted if the credential holder passes through the access point. How this is determined is a local matter. The occupancy count of the zones is adjusted by writing a negative amount to the Adjust_Value property of the exit access zone and the corresponding positive amount to the Adjust_Value property of the entry access zone. If this property is not supported, then the Access Point object behaves as if the value is FALSE.

Property *occupancyCountAdjust* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.occupancyCountAdjust

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 291

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessPoint.occupancyLowerLimitEnforced

According to the BACnet protocol documentation:

Indicates whether the lower occupancy limit of the access controlled zone, for which this object is an exit point, is enforced (TRUE) or not (FALSE). If enforced, authorization shall fail if the access controlled zone's occupancy is lower than or equal to its lower occupancy limit, unless the credential is exempted from this authorization check. When this authorization check fails, the Access_Event property shall be set to DENIED_LOWER_OCCUPANCY_LIMIT.

Property *occupancyLowerLimitEnforced* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.occupancyLowerLimitEnforced

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 295

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessPoint.occupancyUpperLimitEnforced

According to the BACnet protocol documentation:

Indicates whether the upper occupancy limit of the access controlled zone, for which this object is an entry point, is enforced (TRUE) or not (FALSE). If enforced, authorization shall fail if the access controlled zone's occupancy is greater than or equal to its upper occupancy limit, unless the credential is exempted from this authorization check. When this authorization check fails, the Access_Event property shall be set to DENIED_UPPER_OCCUPANCY_LIMIT.

Property *occupancyUpperLimitEnforced* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.occupancyUpperLimitEnforced

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 298

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessPoint.outOfService

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessPoint.priorityForWriting

According to the BACnet protocol documentation:

Provides a priority to be used by the command prioritization mechanism.

Property *priorityForWriting* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.priorityForWriting

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 88

Tag Value

A 32-bit unsigned integer.

accessPoint.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

accessPoint.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

accessPoint.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

accessPoint.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

accessPoint.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessPoint.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *accessPoint* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 33
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessPoint.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

accessPoint.threatLevel

According to the BACnet protocol documentation:

Specifies the current threat level for this Access Point. Zero is the lowest threat level, effectively disabling the threat level check, while 100 is the maximum threat level. If the threat authority of the authenticated credential is lower than the value of this property, then the authorization fails. In this case the Access_Event property shall be set to DENIED_THREAT_LEVEL.

Property *threatLevel* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.threatLevel

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 307

Tag Value

A 32-bit unsigned integer.

accessPoint.transactionNotificationClass

According to the BACnet protocol documentation:

Specifies the instance of the Notification Class object to use for event-notificationdistribution of Access Transaction Events. If this property is not present, then the Notification Class specified by the property Notification_Class shall be used for Access Transaction Events.

Property *transactionNotificationClass* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.transactionNotificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 309

Tag Value

A 32-bit unsigned integer.

accessPoint.verificationTime

According to the BACnet protocol documentation:

Specifies the time, in seconds, to wait for external verification when the Authorization_Mode property has a value of AUTHORIZATION_DELAYED or VERIFICATION_REQUIRED.

Property *verificationTime* of object *accessPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.verificationTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 33
N3:	<objectAddress>
N4:	<property> = 326

Tag Value

A 32-bit unsigned integer.

accessPoint.zoneFrom

According to the BACnet protocol documentation:

Specifies the Access Zone object for which this Access Point object is an exit access controlled point, allowing exit from the zone. This property shall not reference the same Access Zone object as the Zone_To property. If the Access Point is not an exit point from an access controlled zone, then this property shall contain 4194303 in the instance part of the object identifier and in the device instance part of the device identifier, if present.

Property *zoneFrom* of object *accessPoint* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.zoneFrom
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 33
B3:	<objectAddress>
B4:	<property> = 320
Size:	2 elements

Block Elements

Index	Name	Type
0	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.
1	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessPoint.zoneTo

According to the BACnet protocol documentation:

Specifies the Access Zone object for which this Access Point object is an entry access controlled point, allowing entrance to the zone. This property shall not reference the same Access Zone object as the Zone_From property. If the Access Point is not an entry point to an access controlled zone, then this property shall contain 4194303 in the instance part of the object identifier and in the device instance part of the device identifier, if present.

Property *zoneTo* of object *accessPoint* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessPoint-<objectAddress>.zoneTo
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 33
B3:	<objectAddress>
B4:	<property> = 321
Size:	2 elements

Block Elements

Index	Name	Type
0	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.
1	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessRights Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics associated with access rights for physical access control. The Access Rights object is a collection of individual access rule specifications which define privileges for entering and leaving access controlled zones or for accessing other resources or functions. One or many credentials can share this collection of access rules. This object type supports role-based access control models.

Objects of type accessRights have the following properties:

- **accompaniment**
- **description**
- **enable**
- **globalIdentifier**
- **negativeAccessRules**
- **objectIdentifier**
- **objectName**
- **objectType**
- **positiveAccessRules**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**

- tags

accessRights.accompaniment

According to the BACnet protocol documentation:

Specifies that the access rights, which this object represents, may be evaluated successfully only if the original credential, which has this Access Rights object assigned, is accompanied by a second credential that meets the accompaniment criteria and is presented at the same access point. The accompanying credential must also have valid access rights to the Access Point where both credentials are presented. It is a local matter as to whether the accompanying credential is required to be presented by the access user before or after the original credential.

Property *accompaniment* of object *accessRights* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.accompaniment
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 34
B3:	<objectAddress>
B4:	<property> = 252
Size:	2 elements

Block Elements

Index	Name	Type
0	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.
1	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessRights.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

accessRights.enable

According to the BACnet protocol documentation:

Indicates and controls whether (TRUE) or not (FALSE) logging of events is enabled. Logging occurs if and only if Enable is TRUE, Local_Time is on or after Start_Time, and Local_Time is before Stop_Time. If Start_Time contains an unspecified datetime, then it shall be considered equal to 'the start of time'. If Stop_Time contains an unspecified datetime, then it shall be considered equal to 'the end of time'. Log records of type log-status are recorded without regard to the value of the Enable property.

Property *enable* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.enable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 133

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessRights.globalIdentifier

According to the BACnet protocol documentation:

A unique identifier which is used to globally identify the access controlled zone this object represents. This value may be used to identify Access Zone objects in multiple devices that represent the same access controlled zone. If this value is assigned, it shall be unique internetwork-wide and all Access Zone objects in all devices that represent this access controlled zone shall have this value. A value of zero indicates that no global identifier is assigned.

Property *globalIdentifier* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.globalIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 323

Tag Value

A 32-bit unsigned integer.

accessRights.negativeAccessRules

According to the BACnet protocol documentation:

Specifies the negative access rules.

Property *negativeAccessRules* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.negativeAccessRules

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 288

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAccessRule>**.

accessRights.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessRights.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

accessRights.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

accessRights.positiveAccessRules

According to the BACnet protocol documentation:

Specifies the positive access rules.

Property *positiveAccessRules* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.positiveAccessRules

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 302

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAccessRule>**.

accessRights.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

accessRights.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

accessRights.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

accessRights.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

accessRights.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessRights.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *accessRights* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 34
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessRights.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *accessRights* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessRights-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 34
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

accessUser Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics associated with a user of a physical access control system. The Access User object is used to represent an individual person, a group of users, or an asset. Relationships among access users are supported for representation of hierarchical organizations (e.g., companies, departments, or groups of any kind) or for representing ownership of assets.

Objects of type accessUser have the following properties:

- **credentials**
- **description**
- **globalIdentifier**
- **memberOf**
- **members**
- **objectIdentifier**
- **objectName**
- **objectType**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**
- **tags**
- **userExternalIdentifier**
- **userInformationReference**
- **userName**
- **userType**

accessUser.credentials

According to the BACnet protocol documentation:

References all Access Credential objects that represent those credentials which are owned by this access user. Each object referenced shall be an Access Credential object.

Property *credentials* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.credentials

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 265

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectReference>**.

accessUser.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

accessUser.globalIdentifier

According to the BACnet protocol documentation:

A unique identifier which is used to globally identify the access controlled zone this object represents. This value may be used to identify Access Zone objects in multiple devices that represent the same access controlled zone. If this value is assigned, it shall be unique internetwork-wide and all Access Zone objects in all devices that represent this access controlled zone shall have this value. A value of zero indicates that no global identifier is assigned.

Property *globalIdentifier* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.globalIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 323

Tag Value

A 32-bit unsigned integer.

accessUser.memberOf

According to the BACnet protocol documentation:

Indicates those LifeSafetyZone objects of which this LifeSafetyPoint object is considered to be a zone member.

Property *memberOf* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.memberOf

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 159

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectReference>**.

accessUser.members

According to the BACnet protocol documentation:

References the Access User objects that represent the associated access users. Each object referenced shall be an Access User object.

Property *members* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.members

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 286

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectReference>**.

accessUser.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessUser.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

accessUser.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

accessUser.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

accessUser.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

accessUser.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

accessUser.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

accessUser.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessUser.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *accessUser* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 35
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessUser.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property tags of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

accessUser.userExternalIdentifier

According to the BACnet protocol documentation:

Specifies an external identifier associated with the access user. While the content is typically unique, its interpretation is a local matter.

Property *userExternalIdentifier* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.userExternalIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 310

Tag Value

A string value.

accessUser.userInformationReference

According to the BACnet protocol documentation:

Specifies a reference to an external system where additional information of the user can be found. The interpretation of the content is a local matter.

Property *userInformationReference* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.userInformationReference

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 311

Tag Value

A string value.

accessUser.userName

According to the BACnet protocol documentation:

A string of printable characters which specifies the name of the access user. The content is not restricted and can contain multiple lines.

Property *userName* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.userName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 317

Tag Value

A string value.

accessUser.userType

According to the BACnet protocol documentation:

Specifies the access user type this object represents.

Property *userType* of object *accessUser* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessUser-<objectAddress>.userType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 35
N3:	<objectAddress>
N4:	<property> = 318

Tag Value

An integer representing one of the options of enumeration **<BACnetAccessUserType>**.

accessZone Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics associated with a secured geographical zone for which authentication and authorization of a credential takes place to obtain physical access. Entrance to the zone takes place through entry access controlled points while the zone is exited through exit access controlled points. These access controlled points are represented by Access Point objects.

Objects of type accessZone have the following properties:

- **ackedTransitions**
- **adjustValue**
- **alarmValues**
- **credentialsInZone**
- **description**
- **entryPoints**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **exitPoints**
- **globalIdentifier**
- **lastCredentialAdded**
- **lastCredentialAddedTime**
- **lastCredentialRemoved**
- **lastCredentialRemovedTime**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**

- **objectType**
- **occupancyCount**
- **occupancyCountEnable**
- **occupancyLowerLimit**
- **occupancyState**
- **occupancyUpperLimit**
- **outOfService**
- **passbackMode**
- **passbackTimeout**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**

accessZone.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *accessZone* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 36
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessZone.adjustValue

According to the BACnet protocol documentation:

Adjusts the Occupancy_Count property when written. When this property is written and the value of the Occupancy_Count_Enable property is FALSE, then the Adjust_Value property shall be set to zero. If Adjust_Value has never been written or the Occupancy_Count_Enable property is FALSE, then this property shall have a value of zero.

Property *adjustValue* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.adjustValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 176

Tag Value

A 32-bit signed integer.

accessZone.alarmValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-OFFNORMAL event is generated.

Property *alarmValues* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.alarmValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 7

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAccessZoneOccupancyState>**.

accessZone.credentialsInZone

According to the BACnet protocol documentation:

Used to list references to those Access Credential objects that represent credentials assumed to be in this zone. This information may be used to verify whether a specific credential is already in the zone for passback detection purposes. If the zone does not support listing credentials, then this list, if present, shall be empty. It is a local matter as to how this list is updated.

Property *credentialsInZone* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.credentialsInZone

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 266

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectReference>**.

accessZone.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

accessZone.entryPoints

According to the BACnet protocol documentation:

References all Access Point objects that lead into the zone.

Property *entryPoints* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.entryPoints

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 268

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectReference>**.

accessZone.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `accessZone` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessZone.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property `eventAlgorithmInhibitRef` of object `accessZone` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

accessZone.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessZone.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *accessZone* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 36
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessZone.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *accessZone* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 36
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

accessZone.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *accessZone* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 36
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

accessZone.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

accessZone.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *accessZone* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 36
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

accessZone.exitPoints

According to the BACnet protocol documentation:

References all Access Point objects that lead out of the zone.

Property *exitPoints* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.exitPoints

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 269

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectReference>**.

accessZone.globalIdentifier

According to the BACnet protocol documentation:

A unique identifier which is used to globally identify the access controlled zone this object represents. This value may be used to identify Access Zone objects in multiple devices that represent the same access controlled zone. If this value is assigned, it shall be unique internetwork-wide and all Access Zone objects in all devices that represent this access controlled zone shall have this value. A value of zero indicates that no global identifier is assigned.

Property *globalIdentifier* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.globalIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 323

Tag Value

A 32-bit unsigned integer.

accessZone.lastCredentialAdded

According to the BACnet protocol documentation:

Indicates the reference to the Access Credential object which has last been added to the Credentials_In_Zone property. If no credential has been added yet, then this reference shall contain 4194303 in the instance part of the object identifier and in the device instance part of the device identifier, if present. If COV property subscriptions for this property are present, then any update, even one with the same value, is reported by a COV notification.

Property *lastCredentialAdded* of object *accessZone* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.lastCredentialAdded
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 36
B3:	<objectAddress>
B4:	<property> = 277
Size:	2 elements

Block Elements

Index	Name	Type
0	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.
1	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessZone.lastCredentialAddedTime

According to the BACnet protocol documentation:

Indicates the date and time when a reference to an Access Credential object has last been added to the Credentials_In_Zone property. If this property is present, but no credential has yet been added, then this property shall not convey an actual time and shall contain a value of X'FF' in all octets.

Property *lastCredentialAddedTime* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.lastCredentialAddedTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 278

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

accessZone.lastCredentialRemoved

According to the BACnet protocol documentation:

Indicates the reference to the Access Credential object which has last been removed from the Credentials_In_Zone property. If no credential has been removed yet, then this reference shall contain 4194303 in the instance part of the object identifier and in the device instance part of the device identifier, if present. If COV property subscriptions for this property are present, then any update, even one with the same value, is reported by a COV notification.

Property *lastCredentialRemoved* of object *accessZone* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.lastCredentialRemoved
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 36
B3:	<objectAddress>
B4:	<property> = 279
Size:	2 elements

Block Elements

Index	Name	Type
0	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.
1	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessZone.lastCredentialRemovedTime

According to the BACnet protocol documentation:

Indicates the date and time when a reference to an Access Credential object has last been removed from the Credentials_In_Zone property. If this property is present, but no credential has yet been removed, then this property shall not convey an actual time and shall contain a value of X'FF' in all octets.

Property *lastCredentialRemovedTime* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.lastCredentialRemovedTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 280

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

accessZone.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

accessZone.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

accessZone.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accessZone.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

accessZone.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

accessZone.occupancyCount

According to the BACnet protocol documentation:

Used to indicate the actual occupancy count of a zone. If the value of the Occupancy_Count_Enable property is FALSE, then this property shall have a value of zero. The value of the Occupancy_Count property may be adjusted by writing to the Adjust_Value property. The Occupancy_Count property shall be writable when Out_Of_Service is TRUE. When Out_Of_Service becomes FALSE, it is a local matter as to what value this property is set to.

Property *occupancyCount* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.occupancyCount

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 290

Tag Value

A 32-bit unsigned integer.

accessZone.occupancyCountEnable

According to the BACnet protocol documentation:

Indicates whether occupancy counting is enabled (TRUE) or not (FALSE). If this property has a value of FALSE, then the Occupancy_State property shall have a value of DISABLED. When this property changes from FALSE to TRUE it is a local matter as to what value the Occupancy_Count property is set to.

Property *occupancyCountEnable* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.occupancyCountEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 292

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessZone.occupancyLowerLimit

According to the BACnet protocol documentation:

Specifies the occupancy lower limit of the zone. If this property has a value of zero, then there is no lower limit.

Property *occupancyLowerLimit* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.occupancyLowerLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 294

Tag Value

A 32-bit unsigned integer.

accessZone.occupancyState

According to the BACnet protocol documentation:

Reflects the occupancy state of the zone.

Property *occupancyState* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.occupancyState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 296

Tag Value

An integer representing one of the options of enumeration **<BACnetAccessZoneOccupancyState>**.

accessZone.occupancyUpperLimit

According to the BACnet protocol documentation:

Specifies the occupancy upper limit of the zone. If this property has a value of zero, then there is no upper limit. If this value is not zero, it shall be greater than the value of the Occupancy_Lower_Limit, if present.

Property *occupancyUpperLimit* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.occupancyUpperLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 297

Tag Value

A 32-bit unsigned integer.

accessZone.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessZone.passbackMode

According to the BACnet protocol documentation:

Specifies how all Access Point objects that represent entry points to the access controlled zone this object represents shall handle passback violations.

Property *passbackMode* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.passbackMode

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 300

Tag Value

An integer representing one of the options of enumeration **<BACnetAccessPassbackMode>**.

accessZone.passbackTimeout

According to the BACnet protocol documentation:

Specifies the passback timeout in minutes. The timeout is evaluated individually for every credential used to enter the zone. The timeout period for a particular credential begins at the time of successful access to the zone. After the timeout has expired for a particular credential, a passback violation of this credential will no longer be detected. A value of zero or absence of this property indicates passback violations will never time out.

Property *passbackTimeout* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.passbackTimeout

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 301

Tag Value

A 32-bit unsigned integer.

accessZone.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

accessZone.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

accessZone.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

accessZone.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

accessZone.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessZone.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *accessZone* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 36
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accessZone.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property tags of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

accessZone.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

accessZone.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *accessZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accessZone-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 36
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

accumulator Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a device that indicates measurements made by counting pulses. This object maintains precise measurement of input count values, accumulated over time. The accumulation of pulses represents the measured quantity in unsigned integer units. This object is also concerned with the accurate representation of values presented on meter read-outs. This includes the ability to initially set the Present_Value property to the value currently displayed by the meter (as when the meter is installed), and to duplicate the means by which it is advanced, including simulating a modulo-N divider prescaling the actual meter display value.

Objects of type accumulator have the following properties:

- **ackedTransitions**
- **description**
- **deviceType**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultHighLimit**
- **faultLowLimit**
- **highLimit**
- **limitEnable**
- **limitMonitoringInterval**
- **loggingObject**
- **loggingRecord**
- **lowLimit**
- **maxPresValue**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**

- **objectType**
- **outOfService**
- **prescale**
- **presentValue**
- **profileLocation**
- **profileName**
- **propertyList**
- **pulseRate**
- **reliability**
- **reliabilityEvaluationInhibit**
- **scale**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **units**
- **valueBeforeChange**
- **valueChangeTime**
- **valueSet**

accumulator.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *accumulator* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 23
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accumulator.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

accumulator.deviceType

According to the BACnet protocol documentation:

String describing the physical device connected to the analog input. It will typically be used to describe the type of sensor attached to the analog input.

Property *deviceType* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.deviceType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 31

Tag Value

A string value.

accumulator.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accumulator.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property `Event_Algorithm_Inhibit`. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type `BACnetBinaryPV` or `BOOLEAN`.

Property `eventAlgorithmInhibitRef` of object `accumulator` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '<index>)' can be used on array properties.

accumulator.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, `Event_State` shall be `NORMAL`, and the properties `Acked_Transitions`, `Event_Time_Stamps`, and `Event_Message_Texts` shall be equal to their respective initial conditions.

Property `eventDetectionEnable` of object `accumulator` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accumulator.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *accumulator* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 23
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accumulator.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *accumulator* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 23
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

accumulator.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *accumulator* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 23
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

accumulator.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

accumulator.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *accumulator* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 23
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

accumulator.faultHighLimit

According to the BACnet protocol documentation:

Specifies a limit that the Present_Value must exceed before a fault event is generated by the FAULT_OUT_OF_RANGE fault algorithm.

Property *faultHighLimit* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.faultHighLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 388

Tag Value

A 32-bit unsigned integer.

accumulator.faultLowLimit

According to the BACnet protocol documentation:

Specifies a limit that the Present_Value must fall below before a fault event is generated by the FAULT_OUT_OF_RANGE fault algorithm.

Property *faultLowLimit* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.faultLowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 389

Tag Value

A 32-bit unsigned integer.

accumulator.highLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must exceed before an event is generated.

Property *highLimit* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.highLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 45

Tag Value

A 32-bit unsigned integer.

accumulator.limitEnable

According to the BACnet protocol documentation:

Conveys two flags that separately enable and disable reporting of highLimit and lowLimit offnormal events and their return to normal.

Property *limitEnable* of object *accumulator* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.limitEnable
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 23
B3:	<objectAddress>
B4:	<property> = 52
Size:	2 elements

Block Elements

Index	Name	Type
0	lowLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	highLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accumulator.limitMonitoringInterval

According to the BACnet protocol documentation:

Specifies the monitoring period in seconds for determining the value of PulseRate.

Property *limitMonitoringInterval* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.limitMonitoringInterval

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 182

Tag Value

A 32-bit unsigned integer.

accumulator.loggingObject

According to the BACnet protocol documentation:

Indicates the object in the same device as the Accumulator object which, when it acquires loggingRecord data from the Accumulator object, shall cause the Accumulator object to acquire, present and store the data from the underlying system.

Property *loggingObject* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.loggingObject

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 183

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accumulator.loggingRecord

According to the BACnet protocol documentation:

List of values that must be acquired and returned 'atomically' in order to allow proper interpretation of the data.

Property *loggingRecord* of object *accumulator* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.loggingRecord
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 23
B3:	<objectAddress>
B4:	<property> = 184
Size:	4 elements

Block Elements

Index	Name	Type
0	Timestamp	A Date, Null or String (Xml) value representing a BACnet date/time. • A null value means the date/time is empty/not set; • A date/time value representing a precise date and time; • A character string containing an XML with a value of type <BACnetDateTime>.
1	PresentValue	A 32-bit unsigned integer.
2	Accumulated Value	A 32-bit unsigned integer.
3	Accumulator Status	An integer representing one of the options of enumeration <BACnetAccumulatorRecord_AccumulatorStatus> .

accumulator.lowLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must fall below before an event is generated.

Property *lowLimit* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.lowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 59

Tag Value

A 32-bit unsigned integer.

accumulator.maxPresValue

According to the BACnet protocol documentation:

Indicates the highest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *maxPresValue* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.maxPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 65

Tag Value

A 32-bit unsigned integer.

accumulator.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

accumulator.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

accumulator.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

accumulator.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

accumulator.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

accumulator.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accumulator.prescale

According to the BACnet protocol documentation:

*Presents the coefficients that are used for converting the pulse signals generated by the measuring instrument into the value displayed by Present_Value. The conversions are performed using integer arithmetic in such a fashion that no measurement-generated pulse signals are lost in the conversion. These coefficients might simply document a conversion performed prior to the reception of the input pulses by the Accumulator object, or they might actually be used by the Accumulator to convert input pulses into the value displayed by Present_Value. Whichever is done is a local matter. This procedure supports non-integral ratios of measurement pulses to Present_Value. For example, in an electrical metering application, the output of the voltage- and current-measuring systems might be 9000/1200 (scale / voltage*current) pulses per kWh, requiring the Accumulator object to accumulate 2/15 kWh/pulse. With this algorithm such pulses can be accurately accumulated and displayed when the units of Present_Value are KILOWATT_HOURS.*

Property *prescale* of object *accumulator* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.prescale
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 23
B3:	<objectAddress>
B4:	<property> = 185
Size:	2 elements

Block Elements

Index	Name	Type
0	Multiplier	A 32-bit unsigned integer.
1	ModuloDivide	A 32-bit unsigned integer.

accumulator.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit unsigned integer.

accumulator.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

accumulator.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

accumulator.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

accumulator.pulseRate

According to the BACnet protocol documentation:

Indicates the number of input pulses received during the most recent period specified by LimitMonitoringInterval.

Property *pulseRate* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.pulseRate

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 186

Tag Value

A 32-bit unsigned integer.

accumulator.reliability

According to the BACnet protocol documentation:

Indicates whether the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

accumulator.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accumulator.scale

According to the BACnet protocol documentation:

Indicates the conversion factor to be multiplied with the value of the Present_Value property to provide a value in the units indicated by Units.

Property *scale* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.scale

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 187

Tag Value

One of the following data types:

- A 32-bit floating point number.
- A 32-bit signed integer.

accumulator.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *accumulator* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 23
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

accumulator.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

accumulator.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

accumulator.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

accumulator.units

According to the BACnet protocol documentation:

Indicates the measurement units of this object.

Property *units* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.units

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 117

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

accumulator.valueBeforeChange

According to the BACnet protocol documentation:

Indicates the value of the presentValue property just prior to the most recent write to the valueSet or valueBeforeChange properties.

Property *valueBeforeChange* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.valueBeforeChange

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 190

Tag Value

A 32-bit unsigned integer.

accumulator.valueChangeTime

According to the BACnet protocol documentation:

Represents the date and time of the most recent occurrence of a write to the valueBeforeChange or valueSet properties.

Property *valueChangeTime* of object *accumulator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.valueChangeTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 192

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

accumulator.valueSet

According to the BACnet protocol documentation:

Indicates the value of the `presentValue` property after the most recent write to the `valueSet` or `valueBeforeChange` properties.

Property `valueSet` of object `accumulator` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	accumulator-<objectAddress>.valueSet

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 23
N3:	<objectAddress>
N4:	<property> = 191

Tag Value

A 32-bit unsigned integer.

alertEnrollment Object

According to the BACnet protocol documentation:

Defines a standardized object that represents and contains the information required for managing information alerts from a BACnet device. "Information alerts" are interesting notifications that are not related to algorithmic or intrinsic reporting of an object. The Alert Enrollment object allows these alerts to be generated without impacting the `Event_State` of the object to which the alerts are related.

Objects of type `alertEnrollment` have the following properties:

- **ackedTransitions**
- **description**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **notificationClass**
- **notifyType**

- **objectIdentifier**
- **objectName**
- **objectType**
- **presentValue**
- **profileLocation**
- **profileName**
- **propertyList**
- **tags**

alertEnrollment.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *alertEnrollment* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 52
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

alertEnrollment.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *alertEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

alertEnrollment.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *alertEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

alertEnrollment.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property `eventAlgorithmInhibitRef` of object `alertEnrollment` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

alertEnrollment.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property `eventDetectionEnable` of object `alertEnrollment` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

alertEnrollment.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *alertEnrollment* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 52
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

alertEnrollment.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *alertEnrollment* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 52
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

alertEnrollment.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *alertEnrollment* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 52
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

alertEnrollment.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *alertEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

alertEnrollment.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *alertEnrollment* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 52
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

alertEnrollment.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *alertEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

alertEnrollment.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *alertEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

alertEnrollment.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *alertEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

alertEnrollment.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *alertEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

alertEnrollment.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *alertEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

alertEnrollment.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *alertEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

alertEnrollment.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If

a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *alertEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

alertEnrollment.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *alertEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

alertEnrollment.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *alertEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

alertEnrollment.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *alertEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	alertEnrollment-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 52
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

analogInput Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of an analog input.

Objects of type analogInput have the following properties:

- **ackedTransitions**
- **covIncrement**
- **deadband**
- **description**
- **deviceType**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultHighLimit**
- **faultLowLimit**
- **highLimit**
- **interfaceValue**
- **limitEnable**
- **lowLimit**
- **maxPresValue**
- **minPresValue**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**

- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **resolution**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **units**
- **updateInterval**

analogInput.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *analogInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 0
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogInput.covIncrement

According to the BACnet protocol documentation:

Specifies the minimum change in presentValue that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

Property *covIncrement* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.covIncrement

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 22

Tag Value

A 32-bit floating point number.

analogInput.deadband

According to the BACnet protocol documentation:

Specifies a range between the highLimit and lowLimit properties, which the presentValue must remain within for a TO-NORMAL event to be generated.

Property *deadband* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.deadband

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 25

Tag Value

A 32-bit floating point number.

analogInput.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

analogInput.deviceType

According to the BACnet protocol documentation:

String describing the physical device connected to the analog input. It will typically be used to describe the type of sensor attached to the analog input.

Property *deviceType* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.deviceType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 31

Tag Value

A string value.

analogInput.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `analogInput` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogInput.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property `eventAlgorithmInhibitRef` of object `analogInput` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

analogInput.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogInput.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *analogInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 0
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogInput.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *analogInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 0
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

analogInput.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *analogInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 0
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

analogInput.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

analogInput.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *analogInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 0
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

analogInput.faultHighLimit

According to the BACnet protocol documentation:

Specifies a limit that the Present_Value must exceed before a fault event is generated by the FAULT_OUT_OF_RANGE fault algorithm.

Property *faultHighLimit* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.faultHighLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 388

Tag Value

A 32-bit floating point number.

analogInput.faultLowLimit

According to the BACnet protocol documentation:

Specifies a limit that the Present_Value must fall below before a fault event is generated by the FAULT_OUT_OF_RANGE fault algorithm.

Property *faultLowLimit* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.faultLowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 389

Tag Value

A 32-bit floating point number.

analogInput.highLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must exceed before an event is generated.

Property *highLimit* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.highLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 45

Tag Value

A 32-bit floating point number.

analogInput.interfaceValue

According to the BACnet protocol documentation:

Indicates the value, in engineering units, of the physical input. If the BACnet device is not capable of knowing the value of the physical input, then the value of this property shall be NULL.

Property *interfaceValue* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.interfaceValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 387

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.

analogInput.limitEnable

According to the BACnet protocol documentation:

Conveys two flags that separately enable and disable reporting of highLimit and lowLimit offnormal events and their return to normal.

Property *limitEnable* of object *analogInput* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.limitEnable
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 0
B3:	<objectAddress>
B4:	<property> = 52
Size:	2 elements

Block Elements

Index	Name	Type
0	lowLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	highLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogInput.lowLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must fall below before an event is generated.

Property *lowLimit* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.lowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 59

Tag Value

A 32-bit floating point number.

analogInput.maxPresValue

According to the BACnet protocol documentation:

Indicates the highest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *maxPresValue* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.maxPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 65

Tag Value

A 32-bit floating point number.

analogInput.minPresValue

According to the BACnet protocol documentation:

Indicates the lowest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *minPresValue* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.minPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 69

Tag Value

A 32-bit floating point number.

analogInput.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

analogInput.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

analogInput.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

analogInput.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

analogInput.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

analogInput.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogInput.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *analogInput* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit floating point number.

analogInput.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

analogInput.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *fileName* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.fileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

analogInput.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

analogInput.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

analogInput.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogInput.resolution

According to the BACnet protocol documentation:

Indicates the smallest recognizable change in presentValue in engineering units (read-only).

Property *resolution* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.resolution

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 106

Tag Value

A 32-bit floating point number.

analogInput.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *analogInput* can be read and written using a block tag with 4 elements.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 0
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogInput.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

analogInput.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

analogInput.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

analogInput.units

According to the BACnet protocol documentation:

Indicates the measurement units of this object.

Property *units* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.units

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 117

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

analogInput.updateInterval

According to the BACnet protocol documentation:

Indicates the maximum period of time between updates to the presentValue in hundredths of a second when the input is not overridden and not out-of-service.

Property *updateInterval* of object *analogInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogInput-<objectAddress>.updateInterval

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 0
N3:	<objectAddress>
N4:	<property> = 118

Tag Value

A 32-bit unsigned integer.

analogOutput Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of an analog output.

Objects of type analogOutput have the following properties:

- **ackedTransitions**
- **commandTimeArray**
- **covIncrement**
- **currentCommandPriority**
- **deadband**
- **description**
- **deviceType**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **highLimit**
- **interfaceValue**
- **lastCommandTime**
- **limitEnable**
- **lowLimit**
- **maxPresValue**

- **minPresValue**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **resolution**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **units**
- **valueSource**
- **valueSourceArray**

analogOutput.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *analogOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 1
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogOutput.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *analogOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 1
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set;

		• A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
7	elm07	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
8	elm08	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
9	elm09	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
10	elm10	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
11	elm11	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
12	elm12	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

13	elm13	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
14	elm14	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
15	elm15	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

analogOutput.covIncrement

According to the BACnet protocol documentation:

Specifies the minimum change in presentValue that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

Property *covIncrement* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.covIncrement

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 22

Tag Value

A 32-bit floating point number.

analogOutput.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

analogOutput.deadband

According to the BACnet protocol documentation:

Specifies a range between the highLimit and lowLimit properties, which the presentValue must remain within for a TO-NORMAL event to be generated.

Property *deadband* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.deadband

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 25

Tag Value

A 32-bit floating point number.

analogOutput.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

analogOutput.deviceType

According to the BACnet protocol documentation:

String describing the physical device connected to the analog input. It will typically be used to describe the type of sensor attached to the analog input.

Property *deviceType* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.deviceType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 31

Tag Value

A string value.

analogOutput.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogOutput.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

analogOutput.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogOutput.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *analogOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 1
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogOutput.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *analogOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 1
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

analogOutput.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *analogOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 1
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

analogOutput.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

analogOutput.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *analogOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 1
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

analogOutput.highLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must exceed before an event is generated.

Property *highLimit* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.highLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 45

Tag Value

A 32-bit floating point number.

analogOutput.interfaceValue

According to the BACnet protocol documentation:

Indicates the value, in engineering units, of the physical input. If the BACnet device is not capable of knowing the value of the physical input, then the value of this property shall be NULL.

Property *interfaceValue* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.interfaceValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 387

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.

analogOutput.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

analogOutput.limitEnable

According to the BACnet protocol documentation:

Conveys two flags that separately enable and disable reporting of highLimit and lowLimit offnormal events and their return to normal.

Property *limitEnable* of object *analogOutput* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.limitEnable
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 1
B3:	<objectAddress>
B4:	<property> = 52
Size:	2 elements

Block Elements

Index	Name	Type
0	lowLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	highLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogOutput.lowLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must fall below before an event is generated.

Property *lowLimit* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.lowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 59

Tag Value

A 32-bit floating point number.

analogOutput.maxPresValue

According to the BACnet protocol documentation:

Indicates the highest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *maxPresValue* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.maxPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 65

Tag Value

A 32-bit floating point number.

analogOutput.minPresValue

According to the BACnet protocol documentation:

Indicates the lowest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *minPresValue* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.minPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 69

Tag Value

A 32-bit floating point number.

analogOutput.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

analogOutput.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

analogOutput.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

analogOutput.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

analogOutput.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

analogOutput.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogOutput.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *analogOutput* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit floating point number.

analogOutput.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *analogOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 1
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

1 elm01

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

2 elm02

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

3

elm03

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

4

elm04

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

5 elm05

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

6 elm06

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

7 elm07

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

8 elm08

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

9

elm09

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

10

elm10

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

11 elm11

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

12 elm12

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

13 elm13

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

14 elm14

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

15 elm15

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

analogOutput.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

analogOutput.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

analogOutput.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

analogOutput.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

analogOutput.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall

have the value *NO_FAULT_DETECTED* unless *Out_Of_Service* is *TRUE* and an alternate value has been written to the *Reliability* property.

Property *reliabilityEvaluationInhibit* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to *FALSE* and any other value corresponds to *TRUE*.

analogOutput.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A 32-bit floating point number.

analogOutput.resolution

According to the BACnet protocol documentation:

Indicates the smallest recognizable change in presentValue in engineering units (read-only).

Property *resolution* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.resolution

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 106

Tag Value

A 32-bit floating point number.

analogOutput.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *analogOutput* can be read and written using a block tag with 4 elements.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 1
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogOutput.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

analogOutput.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

analogOutput.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

analogOutput.units

According to the BACnet protocol documentation:

Indicates the measurement units of this object.

Property *units* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.units

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 117

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

analogOutput.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *analogOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 1
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

analogOutput.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *analogOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogOutput-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 1
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

analogValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of an analog value. An "analog value" is a control system parameter residing in the memory of the BACnet device.

Objects of type analogValue have the following properties:

- **ackedTransitions**
- **commandTimeArray**
- **covIncrement**
- **currentCommandPriority**
- **deadband**
- **description**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultHighLimit**
- **faultLowLimit**
- **highLimit**
- **lastCommandTime**
- **limitEnable**
- **lowLimit**
- **maxPresValue**
- **minPresValue**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**

- **outOfService**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **resolution**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **units**
- **valueSource**
- **valueSourceArray**

analogValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *analogValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 2
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *analogValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 2
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. - A null value means the timestamp is empty/not set; - A date/time value representing a precise date and time; - An integer defining the timestamp sequence number;

1	elm01	• A character string containing an XML with a value of type <BACnetTimeStamp>. A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
7	elm07	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set;

		• A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
8	elm08	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
9	elm09	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
10	elm10	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
11	elm11	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
12	elm12	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
13	elm13	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

14	elm14	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
15	elm15	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

analogValue.covIncrement

According to the BACnet protocol documentation:

Specifies the minimum change in presentValue that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

Property *covIncrement* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.covIncrement

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 22

Tag Value

A 32-bit floating point number.

analogValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

analogValue.deadband

According to the BACnet protocol documentation:

Specifies a range between the highLimit and lowLimit properties, which the presentValue must remain within for a TO-NORMAL event to be generated.

Property *deadband* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.deadband

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 25

Tag Value

A 32-bit floating point number.

analogValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

analogValue.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogValue.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

analogValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *analogValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 2
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property `eventMessageTexts` of object `analogValue` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 2
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

analogValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property `eventMessageTextsConfig` of object `analogValue` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 2
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

analogValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

analogValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *analogValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 2
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

analogValue.faultHighLimit

According to the BACnet protocol documentation:

Specifies a limit that the Present_Value must exceed before a fault event is generated by the FAULT_OUT_OF_RANGE fault algorithm.

Property *faultHighLimit* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.faultHighLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 388

Tag Value

A 32-bit floating point number.

analogValue.faultLowLimit

According to the BACnet protocol documentation:

Specifies a limit that the Present_Value must fall below before a fault event is generated by the FAULT_OUT_OF_RANGE fault algorithm.

Property *faultLowLimit* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.faultLowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 389

Tag Value

A 32-bit floating point number.

analogValue.highLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must exceed before an event is generated.

Property *highLimit* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.highLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 45

Tag Value

A 32-bit floating point number.

analogValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

analogValue.limitEnable

According to the BACnet protocol documentation:

Conveys two flags that separately enable and disable reporting of highLimit and lowLimit offnormal events and their return to normal.

Property *limitEnable* of object *analogValue* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.limitEnable
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 2
B3:	<objectAddress>
B4:	<property> = 52
Size:	2 elements

Block Elements

Index	Name	Type
0	lowLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	highLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogValue.lowLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must fall below before an event is generated.

Property *lowLimit* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.lowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 59

Tag Value

A 32-bit floating point number.

analogValue.maxPresValue

According to the BACnet protocol documentation:

Indicates the highest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *maxPresValue* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.maxPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 65

Tag Value

A 32-bit floating point number.

analogValue.minPresValue

According to the BACnet protocol documentation:

Indicates the lowest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *minPresValue* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.minPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 69

Tag Value

A 32-bit floating point number.

analogValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

analogValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

analogValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

analogValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

analogValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

analogValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *analogValue* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit floating point number.

analogValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *analogValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 2
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
1	elm01	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
2	elm02	One of the following data types:

analogValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

analogValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

analogValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

analogValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

analogValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A 32-bit floating point number.

analogValue.resolution

According to the BACnet protocol documentation:

Indicates the smallest recognizable change in presentValue in engineering units (read-only).

Property *resolution* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.resolution

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 106

Tag Value

A 32-bit floating point number.

analogValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *analogValue* can be read and written using a block tag with 4 elements.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 2
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

analogValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

analogValue.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

analogValue.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

analogValue.units

According to the BACnet protocol documentation:

Indicates the measurement units of this object.

Property *units* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.units

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 117

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

analogValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *analogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 2
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

analogValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *analogValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	analogValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 2
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

2	elm02	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types:

		• A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

averaging Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a value that is sampled periodically over a specified time interval. The Averaging object records the minimum, maximum and average value over the interval, and makes these values visible as properties of the Averaging object. The sampled value may be the value of any BOOLEAN, INTEGER, Unsigned, Enumerated or REAL property value of any object within the BACnet device in which the object resides. Optionally, the object property to be sampled may exist in a different BACnet device. The Averaging object shall use a "sliding window" technique that maintains a buffer of N samples distributed over the specified interval. Every (time interval/N) seconds a new sample is recorded displacing the oldest sample from the buffer. At this time, the minimum, maximum and average are recalculated. The buffer shall maintain an indication for each sample that permits the average calculation and minimum/maximum algorithm to determine the number of valid samples in the buffer.

Objects of type averaging have the following properties:

- **attemptedSamples**
- **averageValue**
- **description**
- **maximumValue**
- **maximumValueTimestamp**
- **minimumValue**
- **minimumValueTimestamp**
- **objectIdentifier**
- **objectName**
- **objectPropertyReference**
- **objectType**
- **profileLocation**
- **profileName**
- **propertyList**
- **tags**
- **validSamples**
- **varianceValue**
- **windowInterval**
- **windowSamples**

averaging.attemptedSamples

According to the BACnet protocol documentation:

Indicates the number of samples that have been attempted to be collected for the current window.

Property *attemptedSamples* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.attemptedSamples

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 124

Tag Value

A 32-bit unsigned integer.

averaging.averageValue

According to the BACnet protocol documentation:

Reflects the average value contained within the buffer window for the most recent 'WindowSamples' samples.

Property *averageValue* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.averageValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 125

Tag Value

A 32-bit floating point number.

averaging.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

averaging.maximumValue

According to the BACnet protocol documentation:

Reflects the highest value within the buffer window for the most recent 'WindowSamples' samples.

Property *maximumValue* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.maximumValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 135

Tag Value

A 32-bit floating point number.

averaging.maximumValueTimestamp

According to the BACnet protocol documentation:

Indicates the date and time at which the value stored in 'MaximumValue' was sampled.

Property *maximumValueTimestamp* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.maximumValueTimestamp

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 149

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

averaging.minimumValue

According to the BACnet protocol documentation:

The lowest value contained within the buffer window for the most recent 'WindowSamples' samples.

Property *minimumValue* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.minimumValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 136

Tag Value

A 32-bit floating point number.

averaging.minimumValueTimestamp

According to the BACnet protocol documentation:

Indicates the date and time at which the value stored in 'MinimumValue' was sampled.

Property *minimumValueTimestamp* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.minimumValueTimestamp

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 150

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

averaging.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

averaging.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

averaging.objectPropertyReference

According to the BACnet protocol documentation:

Designates the particular object and property referenced by this object.

Property *objectPropertyReference* of object *averaging* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.objectPropertyReference
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 18
B3:	<objectAddress>
B4:	<property> = 78
Size:	4 elements

Block Elements

Index	Name	Type
0	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').
1	PropertyIdentifier	An integer representing one of the options of enumeration <BACnetPropertyIdentifier> .
2	PropertyArrayIndex	A 32-bit unsigned integer. OBS: this is an optional element, a <null> value indicates the element is not present.
3	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.

averaging.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

averaging.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

averaging.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

averaging.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

averaging.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

averaging.validSamples

According to the BACnet protocol documentation:

Indicates the number of samples that have been successfully collected for the current window.

Property *validSamples* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.validSamples

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 146

Tag Value

A 32-bit unsigned integer.

averaging.varianceValue

According to the BACnet protocol documentation:

Reflects the variance value contained within the buffer window for the most recent 'WindowSamples' samples.

Property *varianceValue* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.varianceValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 151

Tag Value

A 32-bit floating point number.

averaging.windowInterval

According to the BACnet protocol documentation:

Indicates the period of time in seconds over which the minimum, maximum and average values are calculated.

Property *windowInterval* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.windowInterval

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 147

Tag Value

A 32-bit unsigned integer.

averaging.windowSamples

According to the BACnet protocol documentation:

Indicates the number of samples to be taken during the period of time specified by the 'WindowInterval' property.

Property *windowSamples* of object *averaging* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	averaging-<objectAddress>.windowSamples

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 18
N3:	<objectAddress>
N4:	<property> = 148

Tag Value

A 32-bit unsigned integer.

binaryInput Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a binary input. A "binary input" is a physical device or hardware input that can be in only one of two distinct states. In this description, those states are referred to as ACTIVE and INACTIVE. A typical use of a binary input is to indicate whether a particular piece of mechanical equipment, such as a fan or pump, is running or idle. The state ACTIVE corresponds to the situation when the equipment is on or running, and INACTIVE corresponds to the situation when the equipment is off or idle.

Objects of type binaryInput have the following properties:

- **ackedTransitions**
- **activeText**
- **alarmValue**
- **changeOfStateCount**
- **changeOfStateTime**
- **description**
- **deviceType**
- **elapsedActiveTime**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **inactiveText**
- **interfaceValue**
- **notificationClass**

- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **polarity**
- **presentValue**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **timeOfActiveTimeReset**
- **timeOfStateCountReset**

binaryInput.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *binaryInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 3
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryInput.activeText

According to the BACnet protocol documentation:

String characterizing the intended effect of the ACTIVE state of the presentValue property from the human operator's viewpoint.

Property *activeText* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.activeText

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 4

Tag Value

A string value.

binaryInput.alarmValue

According to the BACnet protocol documentation:

Specifies the value that the presentValue must have before an event is generated. Use 0 for inactive and 1 for active.

Property *alarmValue* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.alarmValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 6

Tag Value

An integer representing one of the options of enumeration **<BACnetBinaryPV>**.

binaryInput.changeOfStateCount

According to the BACnet protocol documentation:

Represents the number of times that the presentValue property has changed state since the changeOfStateCount property was most recently set to a zero value.

Property *changeOfStateCount* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.changeOfStateCount

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 15

Tag Value

A 32-bit unsigned integer.

binaryInput.changeOfStateTime

According to the BACnet protocol documentation:

Represents the date and time at which the most recent change of state occurred.

Property *changeOfStateTime* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.changeOfStateTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 16

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryInput.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

binaryInput.deviceType

According to the BACnet protocol documentation:

String describing the physical device connected to the analog input. It will typically be used to describe the type of sensor attached to the analog input.

Property *deviceType* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.deviceType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 31

Tag Value

A string value.

binaryInput.elapsedActiveTime

According to the BACnet protocol documentation:

Represents the accumulated number of seconds that the presentValue property has had the value ACTIVE since the elapsedActiveTime property was most recently set to a zero value.

Property *elapsedActiveTime* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.elapsedActiveTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 33

Tag Value

A 32-bit unsigned integer.

binaryInput.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryInput.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

binaryInput.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryInput.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *binaryInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 3
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryInput.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *binaryInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 3
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

binaryInput.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *binaryInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 3
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

binaryInput.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

binaryInput.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *binaryInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 3
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

binaryInput.inactiveText

According to the BACnet protocol documentation:

String characterizing the intended effect of the INACTIVE state of the presentValue property from the human operator's viewpoint.

Property *inactiveText* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.inactiveText

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 46

Tag Value

A string value.

binaryInput.interfaceValue

According to the BACnet protocol documentation:

Indicates the value, in engineering units, of the physical input. If the BACnet device is not capable of knowing the value of the physical input, then the value of this property shall be NULL.

Property *interfaceValue* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.interfaceValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 387

Tag Value

One of the following data types:

- A null/empty value.
- An integer representing one of the options of enumeration **<BACnetBinaryPV>**.

binaryInput.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

binaryInput.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

binaryInput.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

binaryInput.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

binaryInput.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

binaryInput.outOfService

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryInput.polarity

According to the BACnet protocol documentation:

Indicates the relationship between the physical state of the Input and logical state represented by the Present_Value property. Use 0 for normal polarity, 1 for reverse.

Property *polarity* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.polarity

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 84

Tag Value

An integer representing one of the options of enumeration **<BACnetPolarity>**.

binaryInput.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *binaryInput* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

An integer representing one of the options of enumeration **<BACnetBinaryPV>**.

binaryInput.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

binaryInput.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

binaryInput.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

binaryInput.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

binaryInput.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall

have the value *NO_FAULT_DETECTED* unless *Out_Of_Service* is *TRUE* and an alternate value has been written to the *Reliability* property.

Property *reliabilityEvaluationInhibit* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to *FALSE* and any other value corresponds to *TRUE*.

binaryInput.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *binaryInput* can be read and written using a block tag with 4 elements.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 3
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryInput.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

binaryInput.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

binaryInput.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

binaryInput.timeOfActiveTimeReset

According to the BACnet protocol documentation:

Represents the date and time at which the elapsedActiveTime property was most recently set to a zero value.

Property *timeOfActiveTimeReset* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.timeOfActiveTimeReset

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 114

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryInput.timeOfStateCountReset

According to the BACnet protocol documentation:

Represents the date and time at which the changeOfStateCount property was most recently set to a zero value.

Property *timeOfStateCountReset* of object *binaryInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryInput-<objectAddress>.timeOfStateCountReset

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 3
N3:	<objectAddress>
N4:	<property> = 115

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryLightingOutput Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a binary lighting output and includes dedicated functionality specific to lighting control that would otherwise require explicit programming. This object is binary in nature and can be either in a logical on or off state. The object is commanded by writing to the Present_Value property. This property is commandable and uses a priority array to arbitrate between multiple writers to the binary lighting output.

Objects of type binaryLightingOutput have the following properties:

- **ackedTransitions**
- **blinkWarnEnable**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **egressActive**
- **egressTime**
- **elapsedActiveTime**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventTimeStamps**
- **feedbackValue**
- **lastCommandTime**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **polarity**
- **power**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**

- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **statusFlags**
- **strikeCount**
- **tags**
- **timeOfActiveTimeReset**
- **timeOfStrikeCountReset**
- **valueSource**
- **valueSourceArray**

binaryLightingOutput.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *binaryLightingOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 55
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryLightingOutput.blinkWarnEnable

According to the BACnet protocol documentation:

Specifies whether a blink-warn is executed (TRUE) or not (FALSE) when a WARN, WARN_RELINQUISH, or WARN_OFF command is written to the Lighting_Command property or one of the special values is written to the Present_Value. When this property is FALSE and a warn operation is written, a blink-warn notification shall not occur, and the effect of the operation shall occur immediately without an egress delay.

Property *blinkWarnEnable* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.blinkWarnEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 373

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryLightingOutput.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *binaryLightingOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 55
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

binaryLightingOutput.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

binaryLightingOutput.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

binaryLightingOutput.egressActive

According to the BACnet protocol documentation:

Shall be TRUE whenever the Egress_Time for a WARN_RELINQUISH or WARN_OFF lighting operation is in effect and FALSE otherwise.

Property *egressActive* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.egressActive

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 386

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryLightingOutput.egressTime

According to the BACnet protocol documentation:

specifies the egress time in seconds when a WARN_RELINQUISH or WARN_OFF is written to the Lighting_Command property or when the special values -2.0 or -3.0 are written to the Present_Value property. The egress time is the time for which the light level is held at its current level before it is relinquished or set to 0.0%.

Property *egressTime* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.egressTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 377

Tag Value

A 32-bit unsigned integer.

binaryLightingOutput.elapsedActiveTime

According to the BACnet protocol documentation:

Represents the accumulated number of seconds that the presentValue property has had the value ACTIVE since the elapsedActiveTime property was most recently set to a zero value.

Property *elapsedActiveTime* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.elapsedActiveTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 33

Tag Value

A 32-bit unsigned integer.

binaryLightingOutput.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and

the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property `eventDetectionEnable` of object `binaryLightingOutput` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryLightingOutput.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property `eventEnable` of object `binaryLightingOutput` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 55
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryLightingOutput.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *binaryLightingOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 55
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

binaryLightingOutput.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *binaryLightingOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 55
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

binaryLightingOutput.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *binaryLightingOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 55
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

binaryLightingOutput.feedbackValue

According to the BACnet protocol documentation:

Indicates the status of a feedback value from which the presentValue must differ before an event is generated.

Property *feedbackValue* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.feedbackValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 40

Tag Value

An integer representing one of the options of enumeration **<BACnetBinaryLightingPV>**.

binaryLightingOutput.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

binaryLightingOutput.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

binaryLightingOutput.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

binaryLightingOutput.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

binaryLightingOutput.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

binaryLightingOutput.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

binaryLightingOutput.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryLightingOutput.polarity

According to the BACnet protocol documentation:

Indicates the relationship between the physical state of the Input and logical state represented by the Present_Value property. Use 0 for normal polarity, 1 for reverse.

Property *polarity* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.polarity

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 84

Tag Value

An integer representing one of the options of enumeration **<BACnetPolarity>**.

binaryLightingOutput.power

According to the BACnet protocol documentation:

Indicates the nominal power consumption of the load(s) controlled by this object when the light level is 100.0% of the non-normalized range. The units shall be kilowatts.

Property *power* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.power

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 384

Tag Value

A 32-bit floating point number.

binaryLightingOutput.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

An integer representing one of the options of enumeration **<BACnetBinaryLightingPV>**.

binaryLightingOutput.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *binaryLightingOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 55
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

1 elm01

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

2 elm02

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

3 elm03

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

4 elm04

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

5 elm05

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

6 elm06

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

7 elm07

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

8 elm08

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

9 elm09

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

10 elm10

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

11 elm11

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

12 elm12

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

13 elm13

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

14 elm14

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

15 elm15

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryLightingOutput.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If

a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

binaryLightingOutput.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

binaryLightingOutput.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

binaryLightingOutput.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

binaryLightingOutput.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryLightingOutput.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

An integer representing one of the options of enumeration **<BACnetBinaryLightingPV>**.

binaryLightingOutput.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *binaryLightingOutput* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 55
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryLightingOutput.strikeCount

According to the BACnet protocol documentation:

Represents the number of times that the Present_Value property or Feedback_Value property has transitioned from OFF to ON since the date and time indicated by the Time_Of_Strike_Count_Reset property. Whether Present_Value or Feedback_Value is used as the indicator for the calculation of the Strike_Count is a local matter. When this property is set to

zero, the *Time_Of_Strike_Count_Reset* property shall be set to the current date and time. When this property is set to a non-zero value, the value of the *Time_Of_Strike_Count_Reset* property shall not change. If this property is writable, it shall be a local matter whether the property accepts writes of zero only. The *Strike_Count* property shall have a range of 0-65535 or greater.

Property *strikeCount* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.strikeCount

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 391

Tag Value

A 32-bit unsigned integer.

binaryLightingOutput.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

binaryLightingOutput.timeOfActiveTimeReset

According to the BACnet protocol documentation:

Represents the date and time at which the elapsedActiveTime property was most recently set to a zero value.

Property *timeOfActiveTimeReset* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.timeOfActiveTimeReset

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 114

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryLightingOutput.timeOfStrikeCountReset

According to the BACnet protocol documentation:

Indicates the date and time at which the counting of lamp strikes, indicated by the Strike_Count property, has been started.

Property *timeOfStrikeCountReset* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.timeOfStrikeCountReset

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 392

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryLightingOutput.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *binaryLightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 55
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

binaryLightingOutput.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *binaryLightingOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryLightingOutput-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 55
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

binaryOutput Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a binary output. A "binary output" is a physical device or hardware output that can be in only one of two distinct states. In this description, those states are referred to as ACTIVE and INACTIVE. A typical use of a binary output is to switch a particular piece of mechanical equipment, such as a fan or pump, on or off. The state ACTIVE corresponds to the situation when the equipment is on or running, and INACTIVE corresponds to the situation when the equipment is off or idle.

Objects of type binaryOutput have the following properties:

- **ackedTransitions**
- **activeText**
- **changeOfStateCount**
- **changeOfStateTime**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **deviceType**
- **elapsedActiveTime**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **feedbackValue**
- **inactiveText**
- **interfaceValue**
- **lastCommandTime**
- **minimumOffTime**
- **minimumOnTime**
- **notificationClass**
- **notifyType**
- **objectIdentifier**

- **objectName**
- **objectType**
- **outOfService**
- **polarity**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **timeOfActiveTimeReset**
- **timeOfStateCountReset**
- **valueSource**
- **valueSourceArray**

binaryOutput.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *binaryOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 4
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryOutput.activeText

According to the BACnet protocol documentation:

String characterizing the intended effect of the ACTIVE state of the presentValue property from the human operator's viewpoint.

Property *activeText* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.activeText

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 4

Tag Value

A string value.

binaryOutput.changeOfStateCount

According to the BACnet protocol documentation:

Represents the number of times that the presentValue property has changed state since the changeOfStateCount property was most recently set to a zero value.

Property *changeOfStateCount* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.changeOfStateCount

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 15

Tag Value

A 32-bit unsigned integer.

binaryOutput.changeOfStateTime

According to the BACnet protocol documentation:

Represents the date and time at which the most recent change of state occurred.

Property *changeOfStateTime* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.changeOfStateTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 16

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryOutput.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *binaryOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 4
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

binaryOutput.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

binaryOutput.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

binaryOutput.deviceType

According to the BACnet protocol documentation:

String describing the physical device connected to the analog input. It will typically be used to describe the type of sensor attached to the analog input.

Property *deviceType* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.deviceType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 31

Tag Value

A string value.

binaryOutput.elapsedActiveTime

According to the BACnet protocol documentation:

Represents the accumulated number of seconds that the presentValue property has had the value ACTIVE since the elapsedActiveTime property was most recently set to a zero value.

Property *elapsedActiveTime* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.elapsedActiveTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 33

Tag Value

A 32-bit unsigned integer.

binaryOutput.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryOutput.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

binaryOutput.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryOutput.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *binaryOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 4
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryOutput.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *binaryOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 4
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

binaryOutput.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *binaryOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 4
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

binaryOutput.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

binaryOutput.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *binaryOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 4
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

binaryOutput.feedbackValue

According to the BACnet protocol documentation:

Indicates the status of a feedback value from which the presentValue must differ before an event is generated.

Property *feedbackValue* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.feedbackValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 40

Tag Value

An integer representing one of the options of enumeration **<BACnetBinaryPV>**.

binaryOutput.inactiveText

According to the BACnet protocol documentation:

String characterizing the intended effect of the INACTIVE state of the presentValue property from the human operator's viewpoint.

Property *inactiveText* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.inactiveText

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 46

Tag Value

A string value.

binaryOutput.interfaceValue

According to the BACnet protocol documentation:

Indicates the value, in engineering units, of the physical input. If the BACnet device is not capable of knowing the value of the physical input, then the value of this property shall be NULL.

Property *interfaceValue* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.interfaceValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 387

Tag Value

One of the following data types:

- A null/empty value.
- An integer representing one of the options of enumeration **<BACnetBinaryPV>**.

binaryOutput.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

binaryOutput.minimumOffTime

According to the BACnet protocol documentation:

Represents the minimum number of seconds that the presentValue shall remain in the INACTIVE state after a write to the presentValue property causes that property to assume the INACTIVE state.

Property *minimumOffTime* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.minimumOffTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 66

Tag Value

A 32-bit unsigned integer.

binaryOutput.minimumOnTime

According to the BACnet protocol documentation:

Represents the minimum number of seconds that the presentValue shall remain in the ACTIVE state after a write to the presentValue property causes that property to assume the ACTIVE state.

Property *minimumOnTime* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.minimumOnTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 67

Tag Value

A 32-bit unsigned integer.

binaryOutput.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

binaryOutput.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

binaryOutput.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

binaryOutput.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

binaryOutput.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

binaryOutput.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryOutput.polarity

According to the BACnet protocol documentation:

Indicates the relationship between the physical state of the Input and logical state represented by the Present_Value property. Use 0 for normal polarity, 1 for reverse.

Property *polarity* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.polarity

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 84

Tag Value

An integer representing one of the options of enumeration **<BACnetPolarity>**.

binaryOutput.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *binaryOutput* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

An integer representing one of the options of enumeration **<BACnetBinaryPV>**.

binaryOutput.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *binaryOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 4
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>.

- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

1 elm01

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

2 elm02

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.

- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
 - A character string containing an XML with a value of type **<Any>**.
 - A character string containing an XML with a value of type **<BACnetDateTime>**.
- 3 elm03 One of the following data types:
- A null/empty value.
 - A 32-bit floating point number.
 - An unsigned integer representing an enumerated value.
 - A character string containing an XML with a value of type **<Unsigned>**.
 - A character string containing an XML with a value of type **<Boolean>**.
 - A character string containing an XML with a value of type **<INTEGER>**.
 - A character string containing an XML with a value of type **<Double>**.
 - A Date/Time value (only the Time part).
 - A string value.
 - A character string containing an XML with a value of type **<OctetString>**.
 - A character string containing an XML with a value of type **<BitString>**.
 - A character string containing an XML with a value of type **<Date>**.
 - A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
 - A character string containing an XML with a value of type **<Any>**.
 - A character string containing an XML with a value of type **<BACnetDateTime>**.
- 4 elm04 One of the following data types:
- A null/empty value.
 - A 32-bit floating point number.
 - An unsigned integer representing an enumerated value.
 - A character string containing an XML with a value of type **<Unsigned>**.
 - A character string containing an XML with a value of type **<Boolean>**.
 - A character string containing an XML with a value of type **<INTEGER>**.
 - A character string containing an XML with a value of type **<Double>**.
 - A Date/Time value (only the Time part).
 - A string value.
 - A character string containing an XML with a value of type **<OctetString>**.
 - A character string containing an XML with a value of type **<BitString>**.
 - A character string containing an XML with a value of type **<Date>**.

- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

5 elm05

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

6 elm06

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.

		• A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
7	elm07	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
8	elm08	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>.

- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

9

elm09

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

10

elm10

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.

- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

11 elm11

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

12 elm12

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.

- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

13 elm13

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

14 elm14

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.

- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

15 elm15

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryOutput.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

binaryOutput.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *fileName* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.fileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

binaryOutput.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

binaryOutput.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

binaryOutput.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryOutput.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

An integer representing one of the options of enumeration **<BACnetBinaryPV>**.

binaryOutput.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *binaryOutput* can be read and written using a block tag with 4 elements.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 4
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryOutput.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

binaryOutput.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

binaryOutput.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

binaryOutput.timeOfActiveTimeReset

According to the BACnet protocol documentation:

Represents the date and time at which the elapsedActiveTime property was most recently set to a zero value.

Property *timeOfActiveTimeReset* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.timeOfActiveTimeReset

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 114

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryOutput.timeOfStateCountReset

According to the BACnet protocol documentation:

Represents the date and time at which the changeOfStateCount property was most recently set to a zero value.

Property *timeOfStateCountReset* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.timeOfStateCountReset

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 115

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryOutput.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *binaryOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 4
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.

- A character string containing an XML with a value of type **<BACnetAddress>**.

binaryOutput.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *binaryOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryOutput-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 4
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types:

		• A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

binaryValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a binary value. A "binary value" is a control system parameter residing in the memory of the BACnet device. This parameter may assume only one of two distinct states. In this description, those states are referred to as ACTIVE and INACTIVE.

Objects of type binaryValue have the following properties:

- **ackedTransitions**

- **activeText**
- **alarmValue**
- **changeOfStateCount**
- **changeOfStateTime**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **elapsedActiveTime**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **inactiveText**
- **lastCommandTime**
- **minimumOffTime**
- **minimumOnTime**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**

- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **timeOfActiveTimeReset**
- **timeOfStateCountReset**
- **valueSource**
- **valueSourceArray**

binaryValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *binaryValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 5
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryValue.activeText

According to the BACnet protocol documentation:

String characterizing the intended effect of the ACTIVE state of the presentValue property from the human operator's viewpoint.

Property *activeText* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.activeText

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 4

Tag Value

A string value.

binaryValue.alarmValue

According to the BACnet protocol documentation:

Specifies the value that the presentValue must have before an event is generated. Use 0 for inactive and 1 for active.

Property *alarmValue* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.alarmValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 6

Tag Value

An integer representing one of the options of enumeration **<BACnetBinaryPV>**.

binaryValue.changeOfStateCount

According to the BACnet protocol documentation:

Represents the number of times that the presentValue property has changed state since the changeOfStateCount property was most recently set to a zero value.

Property *changeOfStateCount* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.changeOfStateCount

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 15

Tag Value

A 32-bit unsigned integer.

binaryValue.changeOfStateTime

According to the BACnet protocol documentation:

Represents the date and time at which the most recent change of state occurred.

Property *changeOfStateTime* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.changeOfStateTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 16

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *binaryValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 5
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set;

		• A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
7	elm07	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
8	elm08	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

9	elm09	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
10	elm10	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
11	elm11	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
12	elm12	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
13	elm13	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
14	elm14	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
15	elm15	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time;

- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

binaryValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

binaryValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

binaryValue.elapsedActiveTime

According to the BACnet protocol documentation:

Represents the accumulated number of seconds that the presentValue property has had the value ACTIVE since the elapsedActiveTime property was most recently set to a zero value.

Property *elapsedActiveTime* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.elapsedActiveTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 33

Tag Value

A 32-bit unsigned integer.

binaryValue.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryValue.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

binaryValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system

configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property `eventDetectionEnable` of object `binaryValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property `eventEnable` of object `binaryValue` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 5
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *binaryValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 5
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

binaryValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *binaryValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 5
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

binaryValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

binaryValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *binaryValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 5
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

binaryValue.inactiveText

According to the BACnet protocol documentation:

String characterizing the intended effect of the INACTIVE state of the presentValue property from the human operator's viewpoint.

Property *inactiveText* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.inactiveText

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 46

Tag Value

A string value.

binaryValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

binaryValue.minimumOffTime

According to the BACnet protocol documentation:

Represents the minimum number of seconds that the presentValue shall remain in the INACTIVE state after a write to the presentValue property causes that property to assume the INACTIVE state.

Property *minimumOffTime* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.minimumOffTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 66

Tag Value

A 32-bit unsigned integer.

binaryValue.minimumOnTime

According to the BACnet protocol documentation:

Represents the minimum number of seconds that the presentValue shall remain in the ACTIVE state after a write to the presentValue property causes that property to assume the ACTIVE state.

Property *minimumOnTime* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.minimumOnTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 67

Tag Value

A 32-bit unsigned integer.

binaryValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

binaryValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

binaryValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

binaryValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

binaryValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

binaryValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *binaryValue* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

An integer representing one of the options of enumeration **<BACnetBinaryPV>**.

binaryValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *binaryValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 5
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

1 elm01

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

2 elm02

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

3 elm03

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

4 elm04

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

5 elm05

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

6 elm06

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

7 elm07

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

8 elm08

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

9 elm09

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

10 elm10

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

11 elm11

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

12 elm12

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

13 elm13

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

14 elm14

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

15 elm15

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If

a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

binaryValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

binaryValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

binaryValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

binaryValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

An integer representing one of the options of enumeration **<BACnetBinaryPV>**.

binaryValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *binaryValue* can be read and written using a block tag with 4 elements.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 5
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

binaryValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

binaryValue.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

binaryValue.timeDelayNormal

According to the BACnet protocol documentation:

Contains the `pTimeDelayNormal` parameter for the object's event algorithm.

Property `timeDelayNormal` of object `binaryValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

binaryValue.timeOfActiveTimeReset

According to the BACnet protocol documentation:

Represents the date and time at which the `elapsedActiveTime` property was most recently set to a zero value.

Property `timeOfActiveTimeReset` of object `binaryValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.timeOfActiveTimeReset

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 114

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;

- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryValue.timeOfStateCountReset

According to the BACnet protocol documentation:

Represents the date and time at which the changeOfStateCount property was most recently set to a zero value.

Property *timeOfStateCountReset* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.timeOfStateCountReset

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 115

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

binaryValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *binaryValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 5
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

binaryValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *binaryValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	binaryValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 5
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types:

		• A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value.

- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

bitstringValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a BitString Value object to make any kind of bitstring data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

Objects of type bitstringValue have the following properties:

- **ackedTransitions**
- **alarmValues**
- **bitMask**
- **bitText**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **lastCommandTime**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**

- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **valueSource**
- **valueSourceArray**

bitstringValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *bitstringValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 39
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

bitstringValue.alarmValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-OFFNORMAL event is generated.

Property *alarmValues* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.alarmValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 7

Tag Value

A character string containing an XML with a value of type **<SequenceOfBitString>**.

bitstringValue.bitMask

According to the BACnet protocol documentation:

The pBitMask parameter of the object's event algorithm.

Property *bitMask* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.bitMask

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 342

Tag Value

A string value representing a sequence of bits, where each bit is represented by the character '0' or '1' (ex: '01101010').

bitstringValue.bitText

According to the BACnet protocol documentation:

Represents descriptions of all possible bits of the Present_Value. The number of descriptions matches the number of bits in the Present_Value property.

Property *bitText* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.bitText

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 343

Tag Value

A character string containing an XML with a value of type **<SequenceOfCharacterString>**.

bitstringValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *bitstringValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 39
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

bitstringValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

bitstringValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

bitstringValue.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `bitstringValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

bitstringValue.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property `eventAlgorithmInhibitRef` of object `bitstringValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

bitstringValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

bitstringValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property `eventEnable` of object `bitstringValue` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 39
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

bitstringValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property `eventMessageTexts` of object `bitstringValue` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 39
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

bitstringValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *bitstringValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 39
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

bitstringValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

bitstringValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *bitstringValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 39
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

bitstringValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

bitstringValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

bitstringValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

bitstringValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

bitstringValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

bitstringValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

bitstringValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

bitstringValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A string value representing a sequence of bits, where each bit is represented by the character '0' or '1' (ex: '01101010').

bitstringValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *bitstringValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 39
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
1	elm01	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
2	elm02	One of the following data types:

bitstringValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

bitstringValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

bitstringValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

bitstringValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

bitstringValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

bitstringValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A string value representing a sequence of bits, where each bit is represented by the character '0' or '1' (ex: '01101010').

bitstringValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *bitstringValue* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 39
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

bitstringValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

bitstringValue.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

bitstringValue.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

bitstringValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *bitstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 39
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

bitstringValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *bitstringValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	bitstringValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 39
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

calendar Object

According to the BACnet protocol documentation:

Defines a standardized object used to describe a list of calendar dates, which might be thought of as "holidays", "special events", or simply as a list of dates.

Objects of type calendar have the following properties:

- **dateList**
- **description**
- **objectIdentifier**
- **objectName**
- **objectType**
- **presentValue**
- **profileLocation**
- **profileName**
- **propertyList**
- **tags**

calendar.dateList

According to the BACnet protocol documentation:

A list of dates, date ranges or month/week-of-month/day-of-week specifications.

Property *dateList* of object *calendar* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	calendar-<objectAddress>.dateList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 6
N3:	<objectAddress>
N4:	<property> = 23

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetCalendarEntry>**.

calendar.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *calendar* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	calendar-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 6
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

calendar.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *calendar* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	calendar-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 6
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

calendar.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *calendar* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	calendar-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 6
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

calendar.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *calendar* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	calendar-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 6
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

calendar.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *calendar* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	calendar-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 6
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

calendar.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *calendar* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	calendar-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 6
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

calendar.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *calendar* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	calendar-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 6
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

calendar.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *calendar* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	calendar-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 6
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

calendar.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property tags of object *calendar* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	calendar-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 6
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

channel Object

According to the BACnet protocol documentation:

defines a standardized object used to forward a single received value to a collection of object properties. The collection of object properties may include any combination of object types, as well as properties of different data types. Each Channel object is associated with a single logical "channel" in the range 0..65535. Multiple Channel object instances may be associated with a given channel number. Each Channel object may be a member of zero or more "control groups" to facilitate writing to Channel objects with the WriteGroup service. The Channel object is intended for value distribution and does not maintain a state. Therefore, it does not act on its own and does not contain a priority array. When the Present_Value property of this object is written by the WriteProperty, WritePropertyMultiple, or WriteGroup services, and a 'Priority' is provided in the write, this object shall use this same priority to command the referenced properties.

Objects of type channel have the following properties:

- **ackedTransitions**
- **allowGroupDelayInhibit**
- **channelNumber**
- **controlGroups**
- **description**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**

- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **executionDelay**
- **lastPriority**
- **listOfObjectPropertyReferences**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**
- **tags**
- **valueSource**
- **writeStatus**

channel.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *channel* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 53
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

channel.allowGroupDelayInhibit

According to the BACnet protocol documentation:

Indicates whether WriteGroup service writes to this object, that specify 'Inhibit Delay'=TRUE, may override any execution delay specified in this object. Execution_Delay shall always occur as the result of WriteProperty or WritePropertyMultiple. In the case of WriteGroup, Execution_Delay shall always occur unless the WriteGroup service parameter 'Inhibit Delay' is TRUE, and the Channel object property Allow_Group_Delay_Inhibit is present and has the value TRUE.

Property *allowGroupDelayInhibit* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.allowGroupDelayInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 365

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

channel.channelNumber

According to the BACnet protocol documentation:

Indicates the logical channel number that this Channel object is associated with when the Channel object Present_Value is written to using the WriteGroup service.

Property *channelNumber* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.channelNumber

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 366

Tag Value

A 16-bit integer value ranging from 0 to 65535.

channel.controlGroups

According to the BACnet protocol documentation:

Indicates those logical control groups of which this Channel object is a member. This array shall contain at least one entry. Unused array slots shall contain the value zero, and control group zero shall mean "no assignment." Control_Groups is required to be writable, and it shall be permitted to configure the membership of the Channel object in arbitrary groups by writing the control group numbers into this array in any order, up to the maximum number of simultaneous groups supported by the Channel object. Duplicate entries specifying the same group number shall be permitted. The maximum size of the Control_Groups array shall be a local matter.

Property *controlGroups* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.controlGroups

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 367

Tag Value

A character string containing an XML with a value of type `<SequenceOfUnsigned32>`.

channel.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

channel.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

channel.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *channel* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 53
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

channel.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *channel* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 53
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

channel.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *channel* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 53
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

channel.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

channel.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *channel* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 53
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

channel.executionDelay

According to the BACnet protocol documentation:

Indicates an execution delay in milliseconds for each value to be written in the List_Of_Object_Property_References when the Channel object's Present_Value is written. A value of zero indicates no delay. A non-zero execution delay value shall cause a delay, by that many milliseconds, in the writing to the corresponding referenced value. The resolution of Execution_Delay

shall be a local matter. If present, the Execution_Delay property shall be writable. All delay periods shall "start" at the same time. So, a write of A, B(delay 100), C, D(delay 200) shall immediately write A and C, but delay the writing of B by 100 milliseconds and D by 200 milliseconds. Multiple delayed values shall execute their corresponding delays in parallel.

Property *executionDelay* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.executionDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 368

Tag Value

A character string containing an XML with a value of type **<SequenceOfUnsigned>**.

channel.lastPriority

According to the BACnet protocol documentation:

Conveys the priority at which the Present_Value was most recently written (1..16). If an attempt was made to write to the Present_Value without the 'Priority' parameter, a default priority of 16 (the lowest priority) shall be assumed. The initial value of Last_Priority shall be 16.

Property *lastPriority* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.lastPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 369

Tag Value

A 32-bit unsigned integer.

channel.listOfObjectPropertyReferences

According to the BACnet protocol documentation:

Specifies the Object and Property identifiers of the properties to be written with specific values at specific times on specific days.

Property *listOfObjectPropertyReferences* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.listOfObjectPropertyReferences

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 54

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectPropertyReference>**.

channel.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

channel.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

channel.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

channel.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

channel.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

channel.outOfService

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

channel.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<BACnetLightingCommand>**.

channel.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

channel.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

channel.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

channel.reliability

According to the BACnet protocol documentation:

Indicates whether the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

channel.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

channel.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *channel* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 53
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

channel.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

channel.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

channel.writeStatus

According to the BACnet protocol documentation:

Set to IDLE initially. This property shall be set to IN_PROGRESS when a value is written to the Present_Value property indicating that the Channel object has begun processing the List_Of_Object_Property_References.

Property *writeStatus* of object *channel* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	channel-<objectAddress>.writeStatus

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 53
N3:	<objectAddress>
N4:	<property> = 370

Tag Value

An integer representing one of the options of enumeration **<BACnetWriteStatus>**.

characterstringValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a CharacterString Value object to make any kind of character string data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

Objects of type characterstringValue have the following properties:

- **ackedTransitions**
- **alarmValues**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultValues**
- **lastCommandTime**

- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **valueSource**
- **valueSourceArray**

characterstringValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *characterstringValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 40
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

characterstringValue.alarmValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-OFFNORMAL event is generated.

Property *alarmValues* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.alarmValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 7

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetOptionalCharacterString>**.

characterstringValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *characterstringValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 40
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

characterstringValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

characterstringValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

characterstringValue.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `characterstringValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

characterstringValue.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property `eventAlgorithmInhibitRef` of object `characterstringValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

characterstringValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

characterstringValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property `eventEnable` of object `characterstringValue` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 40
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

characterstringValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property `eventMessageTexts` of object `characterstringValue` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 40
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

characterstringValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *characterstringValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 40
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

characterstringValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

characterstringValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *characterstringValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 40
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

characterstringValue.faultValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-FAULT event is generated.

Property *faultValues* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.faultValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 39

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetOptionalCharacterString>**.

characterstringValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

characterstringValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

characterstringValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

characterstringValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

characterstringValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

characterstringValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

characterstringValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

characterstringValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A string value.

characterstringValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *characterstringValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 40
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

1 elm01

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

2 elm02

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

3 elm03

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

4 elm04

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

5 elm05

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

6 elm06

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

7 elm07

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

8 elm08

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

9 elm09

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

10 elm10

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

11 elm11

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

12 elm12

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

13 elm13

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

14 elm14

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.

- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

15 elm15

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

characterstringValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If

a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

characterstringValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

characterstringValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

characterstringValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

characterstringValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

characterstringValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A string value.

characterstringValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *characterstringValue* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 40
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

characterstringValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

characterstringValue.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

characterstringValue.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

characterstringValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *characterstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 40
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

characterstringValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *characterstringValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	characterstringValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 40
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

command Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a multi-action command procedure. A Command object is used to write a set of values to a group of object properties, based on the "action code" that is written to the Present_Value of the Command object. Whenever the Present_Value property of the Command object is written to, it triggers the Command object to take a set of actions that change the values of a set of other objects' properties.

Objects of type command have the following properties:

- **ackedTransitions**
- **action**
- **actionText**
- **allWritesSuccessful**
- **description**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **inProcess**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **presentValue**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**
- **tags**
- **valueSource**

command.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *command* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 7
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

command.action

According to the BACnet protocol documentation:

Specifies an array of 'action lists'.

Property *action* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.action

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 2

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetActionList>**.

command.actionText

According to the BACnet protocol documentation:

Used to indicate a text string description for each of the possible values of the presentValue property.

Property *actionText* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.actionText

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 3

Tag Value

A character string containing an XML with a value of type **<SequenceOfCharacterString>**.

command.allWritesSuccessful

According to the BACnet protocol documentation:

Indicates the success or failure of the sequence of actions that are triggered when the presentValue property is written to.

Property *allWritesSuccessful* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.allWritesSuccessful

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 9

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

command.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

command.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

command.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *command* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 7
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

command.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *command* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 7
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

command.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *command* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 7
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

command.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

command.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *command* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 7
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

command.inProcess

According to the BACnet protocol documentation:

Set to *TRUE* when a value is written to the *presentValue* property. This *TRUE* value indicates that the Command object has begun processing one of a set of action sequences. Once all of the writes have been attempted by the Command object, the *inProcess* property is set back to *FALSE*.

Property *inProcess* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.inProcess

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 47

Tag Value

An integer value, where 0 corresponds to *FALSE* and any other value corresponds to *TRUE*.

command.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

command.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

command.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

command.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

command.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

command.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit unsigned integer.

command.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

command.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

command.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

command.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

command.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

command.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *command* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 7
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

command.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

command.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *command* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	command-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 7
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

credentialDataInput Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a process that provides authentication factors read by a physical device. An authentication factor is a data element of a credential that is a unique digital identifier used to verify the identity of a credential. A credential may have multiple authentication factors. Examples of physical devices that may be represented by this object type are card readers, keypads, biometric readers, etc.

Objects of type credentialDataInput have the following properties:

- **ackedTransitions**
- **description**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**
- **supportedFormatClasses**
- **supportedFormats**

- tags
- updateTime

credentialDataInput.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *credentialDataInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 37
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

credentialDataInput.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

credentialDataInput.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

credentialDataInput.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *credentialDataInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 37
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

credentialDataInput.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *credentialDataInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 37
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

credentialDataInput.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *credentialDataInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 37
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

credentialDataInput.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

credentialDataInput.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *credentialDataInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 37
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

credentialDataInput.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

credentialDataInput.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

credentialDataInput.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

credentialDataInput.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

credentialDataInput.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

credentialDataInput.outOfService

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

credentialDataInput.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *credentialDataInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.presentValue
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 37
B3:	<objectAddress>
B4:	<property> = 85
Size:	3 elements

Block Elements

Index	Name	Type
0	FormatType	An integer representing one of the options of enumeration <BACnetAuthenticationFactorType> .
1	FormatClass	A 32-bit unsigned integer.
2	Value	A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

credentialDataInput.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

credentialDataInput.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

credentialDataInput.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

credentialDataInput.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

credentialDataInput.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

credentialDataInput.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *credentialDataInput* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 37
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

credentialDataInput.supportedFormatClasses

According to the BACnet protocol documentation:

Specifies the values that the Format-Class field of the Present_Value may take on. The value of the ith element of this array shall be used when an authentication factor is read that is of the format defined in the ith element of the Supported_Formats array. This property is used to distinguish between multiple different supported authentication factor formats, used on a site, of which two or more use the same authentication factor format type and may have colliding value ranges. A value of zero is used as the default where no differentiation is required. Otherwise, the value is site specific and can be any non-zero value.

Property *supportedFormatClasses* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.supportedFormatClasses

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 305

Tag Value

A character string containing an XML with a value of type **<SequenceOfUnsigned>**.

credentialDataInput.supportedFormats

According to the BACnet protocol documentation:

Specifies which authentication factor formats are supported by this object.

Property *supportedFormats* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.supportedFormats

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 304

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAuthenticationFactorFormat>**.

credentialDataInput.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

credentialDataInput.updateTime

According to the BACnet protocol documentation:

Indicates the date and time of the last transition of the timer state. If a transition of the timer state has never occurred, then this property shall take on the unspecified datetime value.

Property *updateTime* of object *credentialDataInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	credentialDataInput-<objectAddress>.updateTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 37
N3:	<objectAddress>
N4:	<property> = 189

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

dateValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a Date Value object to make any kind of date data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client. A Date Value object is

used to represent a single day. In contrast, the Date Pattern Value object can be used to represent multiple recurring dates. Date Value objects may optionally support intrinsic reporting to facilitate the reporting of fault conditions.

Objects of type `dateValue` have the following properties:

- **ackedTransitions**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **lastCommandTime**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **statusFlags**
- **tags**
- **valueSource**
- **valueSourceArray**

dateValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *dateValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 42
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

dateValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *dateValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 42
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

dateValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

dateValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

dateValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

dateValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *dateValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 42
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

dateValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *dateValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 42
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

dateValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *dateValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 42
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

dateValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

dateValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *dateValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 42
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

dateValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

dateValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

dateValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

dateValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

dateValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

dateValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

dateValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

dateValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A Date/Time value (only the Date part).

dateValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *dateValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 42
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
1	elm01	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
2	elm02	One of the following data types:

dateValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

dateValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

dateValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

dateValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

dateValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

dateValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A Date/Time value (only the Date part).

dateValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *dateValue* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 42
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

dateValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

dateValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *dateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 42
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

dateValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *dateValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	dateValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 42
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

datepatternValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a Date Pattern Value object to make any kind of date data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client. Date Pattern objects can be used to represent multiple recurring dates based on rules defined by the pattern of individual fields of the date, some of which can be special values like "even months", or "don't care", which matches any value in that field. Examples of possibilities would be: "every Thursday in May of any year", or "every day in May 2009". Date Pattern Value objects may optionally support intrinsic reporting to facilitate the reporting of fault conditions.

Objects of type `datepatternValue` have the following properties:

- **ackedTransitions**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **lastCommandTime**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **statusFlags**
- **tags**

- **valueSource**
- **valueSourceArray**

datepatternValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *datepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 41
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datepatternValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *datepatternValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 41
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

datepatternValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

datepatternValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

datepatternValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datepatternValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *datepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 41
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datepatternValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *datepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 41
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

datepatternValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *datepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 41
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

datepatternValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

datepatternValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *datepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 41
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

datepatternValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

datepatternValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

datepatternValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

datepatternValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

datepatternValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

datepatternValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

datepatternValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datepatternValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A Date/Time value (only the Date part).

datepatternValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *datepatternValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 41
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
1	elm01	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
2	elm02	One of the following data types:

datepatternValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

datepatternValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

datepatternValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

datepatternValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

datepatternValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datepatternValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A Date/Time value (only the Date part).

datepatternValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *datepatternValue* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 41
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datepatternValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

datepatternValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *datepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 41
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

datepatternValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *datepatternValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datepatternValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 41
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

datetimeValue Object

According to the BACnet protocol documentation:

In a BACnet device. A BACnet device can use a DateTime Value object to make any kind of datetime data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client. A DateTime Value object is used to represent a single moment in time. In contrast, the DateTime Pattern Value object can be used to represent multiple recurring dates and times. DateTime Value objects may optionally support intrinsic reporting to facilitate the reporting of fault conditions.

Objects of type datetimeValue have the following properties:

- **ackedTransitions**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **isUtc**
- **lastCommandTime**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **statusFlags**
- **tags**

- **valueSource**
- **valueSourceArray**

datetimeValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *datetimeValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 44
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimeValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *datetimeValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 44
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

datetimeValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

datetimeValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

datetimeValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimeValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *datetimeValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 44
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimeValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *datetimeValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 44
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

datetimeValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *datetimeValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 44
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

datetimeValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

datetimeValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *datetimeValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 44
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

datetimeValue.isUtc

According to the BACnet protocol documentation:

Indicates whether the Present_Value property indicates a UTC date and time (when TRUE) or a local date and time (when FALSE). If this property is absent, the Present_Value shall be a local date and time.

Property *isUtc* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.isUtc

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 344

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimeValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

datetimeValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

datetimeValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

datetimeValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

datetimeValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

datetimeValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

datetimeValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimeValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

datetimeValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *datetimeValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 44
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

1 elm01

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

2 elm02

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

3

elm03

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

4

elm04

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

5 elm05

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

6 elm06

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

7

elm07

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

8

elm08

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

9

elm09

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

10

elm10

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

11 elm11

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

12 elm12

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

13 elm13

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

14 elm14

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

15 elm15

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

datetimeValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

datetimeValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

datetimeValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

datetimeValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

datetimeValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimeValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

datetimeValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *datetimeValue* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 44
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimeValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

datetimeValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *datetimeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 44
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

datetimeValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *datetimeValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimeValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 44
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

datetimepatternValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a DateTime Pattern Value object to make any kind of datetime data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client. DateTime Pattern objects can be used to represent multiple recurring dates and times based on rules defined by the pattern of individual fields of the date and time, some of which can be special values like "even months", or "don't care", which matches any value in that field. Examples of possibilities would be: "11:00 every Thursday in any June", or "every day in May 2009". DateTime Pattern Value objects may optionally support intrinsic reporting to facilitate the reporting of fault conditions.

Objects of type datetimepatternValue have the following properties:

- **ackedTransitions**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **isUtc**
- **lastCommandTime**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**

- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **statusFlags**
- **tags**
- **valueSource**
- **valueSourceArray**

datetimepatternValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *datetimepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 43
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimepatternValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *datetimepatternValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 43
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

datetimepatternValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

datetimepatternValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

datetimepatternValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimepatternValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *datetimepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 43
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimepatternValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *datetimepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 43
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

datetimepatternValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *datetimepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 43
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

datetimepatternValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

datetimepatternValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *datetimepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 43
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

datetimepatternValue.isUtc

According to the BACnet protocol documentation:

Indicates whether the Present_Value property indicates a UTC date and time (when TRUE) or a local date and time (when FALSE). If this property is absent, the Present_Value shall be a local date and time.

Property *isUtc* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.isUtc

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 344

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimepatternValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

datetimepatternValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

datetimepatternValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

datetimepatternValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

datetimepatternValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

datetimepatternValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

datetimepatternValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimepatternValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

datetimepatternValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *datetimepatternValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 43
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

1 elm01

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

2 elm02

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

3 elm03

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

4 elm04

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

5 elm05

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

6 elm06

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

7 elm07

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

8 elm08

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

9

elm09

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

10

elm10

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

11 elm11

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

12 elm12

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

13 elm13

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

14 elm14

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.

- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

15 elm15

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

datetimepatternValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

datetimepatternValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

datetimepatternValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

datetimepatternValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

datetimepatternValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimepatternValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

datetimepatternValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *datetimepatternValue* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 43
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

datetimepatternValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

datetimepatternValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *datetimepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 43
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

datetimepatternValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *datetimepatternValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	datetimepatternValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 43
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

device Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a BACnet device. There shall be exactly one Device object in each BACnet device. A Device object is referenced by its Object_Identifier property, which is not only unique to the BACnet device that maintains this object but is also unique throughout the BACnet internetwork.

Objects of type device have the following properties:

- **ackedTransitions**
- **activeCovMultipleSubscriptions**
- **activeCovSubscriptions**
- **activeVtSessions**
- **alignIntervals**
- **apduSegmentTimeout**
- **apduTimeout**
- **applicationSoftwareVersion**
- **backupAndRestoreState**
- **backupFailureTimeout**
- **backupPreparationTime**
- **configurationFiles**
- **databaseRevision**
- **daylightSavingsStatus**
- **deployedProfileLocation**
- **description**
- **deviceAddressBinding**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **firmwareRevision**
- **intervalOffset**
- **lastRestartReason**

- **lastRestoreTime**
- **localDate**
- **localTime**
- **location**
- **maxApduLengthAccepted**
- **maxInfoFrames**
- **maxMaster**
- **maxSegmentsAccepted**
- **modelName**
- **notificationClass**
- **notifyType**
- **numberOfApduRetries**
- **objectIdentifier**
- **objectList**
- **objectName**
- **objectType**
- **profileLocation**
- **profileName**
- **propertyList**
- **protocolObjectTypesSupported**
- **protocolRevision**
- **protocolServicesSupported**
- **protocolVersion**
- **reliability**
- **reliabilityEvaluationInhibit**
- **restartNotificationRecipients**
- **restoreCompletionTime**
- **restorePreparationTime**
- **segmentationSupported**
- **serialNumber**
- **statusFlags**
- **structuredObjectList**

- **systemStatus**
- **tags**
- **timeOfDeviceRestart**
- **timeSynchronizationInterval**
- **timeSynchronizationRecipients**
- **utcOffset**
- **utcTimeSynchronizationRecipients**
- **vendorIdentifier**
- **vendorName**
- **vtClassesSupported**

device.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *device* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 8
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

device.activeCovMultipleSubscriptions

According to the BACnet protocol documentation:

Provides a network-visible indication of those COV-multiple subscriptions that are active at any given time. Each list entry constitutes a COV-multiple context and consists of a Recipient, an Issue Confirmed Notifications flag, a Time Remaining value, a Maximum Notification Delay timeout, and a list of monitored objects. For each monitored object, a list of COV Specifications is present with each containing a Monitored Property reference, an optional COV Increment, and a Timestamped flag. Only one COV-multiple context shall be present in this property for a given Recipient and form of notification indicated by the Issue Confirmed Notifications flag. A Recipient is identified by the BACnet address of the COV-client and the Subscriber Process Identifier.

Property *activeCovMultipleSubscriptions* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.activeCovMultipleSubscriptions

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 481

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetCOVMultipleSubscription>**.

device.activeCovSubscriptions

According to the BACnet protocol documentation:

Provides a network-visible indication of those COV subscriptions that are active at any given time. Whenever a COV Subscription is created with the SubscribeCOV or SubscribeCOVProperty service, a new entry is added to the Active_COV_Subscriptions list. Similarly, whenever a COV Subscription is terminated, the corresponding entry shall be removed from the Active_COV_Subscriptions list.

Property *activeCovSubscriptions* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.activeCovSubscriptions

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 152

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetCOVSubscription>**.

device.activeVtSessions

According to the BACnet protocol documentation:

Provides a network-visible indication of those virtual terminal sessions (VT-Sessions) that are active at any given time. Whenever a virtual terminal session is created with the VT-Open service, a new entry is added to the Active_VT_Sessions list. Similarly, whenever a VT-session is terminated, the corresponding entry shall be removed from the Active_VT_Sessions list.

Property *activeVtSessions* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.activeVtSessions

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 5

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetVTSession>**.

device.alignIntervals

According to the BACnet protocol documentation:

Specifies whether (TRUE) or not (FALSE) clock-aligned periodic time synchronizations is enabled.

Property *alignIntervals* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.alignIntervals

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 193

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

device.apduSegmentTimeout

According to the BACnet protocol documentation:

Indicates the amount of time in milliseconds between retransmission of an APDU segment.

Property *apduSegmentTimeout* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.apduSegmentTimeout

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 10

Tag Value

A 32-bit unsigned integer.

device.apduTimeout

According to the BACnet protocol documentation:

Indicates the amount of time in milliseconds between retransmissions of an APDU requiring acknowledgement for which no acknowledgement has been received.

Property *apduTimeout* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.apduTimeout

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 11

Tag Value

A 32-bit unsigned integer.

device.applicationSoftwareVersion

According to the BACnet protocol documentation:

Identifies the version of application software installed in the machine.

Property *applicationSoftwareVersion* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.applicationSoftwareVersion

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 12

Tag Value

A string value.

device.backupAndRestoreState

According to the BACnet protocol documentation:

Indicates a server device's backup and restore state.

Property *backupAndRestoreState* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.backupAndRestoreState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 338

Tag Value

An integer representing one of the options of enumeration **<BACnetBackupState>**.

device.backupFailureTimeout

According to the BACnet protocol documentation:

The time, in seconds, that the device being backed up or restored must wait before unilaterally ending the backup or restore procedure.

Property *backupFailureTimeout* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.backupFailureTimeout

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 153

Tag Value

A 16-bit integer value ranging from 0 to 65535.

device.backupPreparationTime

According to the BACnet protocol documentation:

Indicates the amount of time in seconds that the device might remain unresponsive after the sending of a ReinitializeDevice-ACK at the start of a backup procedure. The device that initiated the backup shall either wait the period of time specified by this property or be prepared to encounter communication timeouts during this period.

Property *backupPreparationTime* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.backupPreparationTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 339

Tag Value

A 16-bit integer value ranging from 0 to 65535.

device.configurationFiles

According to the BACnet protocol documentation:

Entries in this array identify the files within the device that define the device's image that can be backed up. The content of this property is only required to be valid during the backup procedure.

Property *configurationFiles* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.configurationFiles

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 154

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetObjectIdentifier>**.

device.databaseRevision

According to the BACnet protocol documentation:

The 'logical revision number' for the device's database.

Property *databaseRevision* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.databaseRevision

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 155

Tag Value

A 32-bit unsigned integer.

device.daylightSavingsStatus

According to the BACnet protocol documentation:

Indicates whether daylight savings time is in effect (TRUE) or not (FALSE) at the BACnet Device's location.

Property *daylightSavingsStatus* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.daylightSavingsStatus

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 24

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

device.deployedProfileLocation

According to the BACnet protocol documentation:

Represents the URI of the location of an xdd file. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. The referenced xdd file contains additional information about the deployed device. It is intended to be used as a supplement to the information referenced by the Profile_Location property. If present, this property shall be writable and shall, at a minimum, support storage of strings with an encoded length up to 255 octets.

Property *deployedProfileLocation* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.deployedProfileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 484

Tag Value

A string value.

device.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

device.deviceAddressBinding

According to the BACnet protocol documentation:

List of object identifiers which identify the actual device addresses that will be used when the remote device must be accessed via a BACnet service request.

Property *deviceAddressBinding* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.deviceAddressBinding

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 30

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAddressBinding>**.

device.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

device.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *device* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 8
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

device.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *device* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 8
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

device.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *device* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 8
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

device.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

device.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *device* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 8
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

device.firmwareRevision

According to the BACnet protocol documentation:

Assigned by the vendor to represent the level of firmware installed in the BACnet device.

Property *firmwareRevision* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.firmwareRevision

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 44

Tag Value

A string value.

device.intervalOffset

According to the BACnet protocol documentation:

Specifies the offset in minutes from the beginning of the period specified for time synchronization until the actual time synchronization requests are sent.

Property *intervalOffset* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.intervalOffset

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 195

Tag Value

A 32-bit unsigned integer.

device.lastRestartReason

According to the BACnet protocol documentation:

Indicates the reason for the last device restart.

Property *lastRestartReason* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.lastRestartReason

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 196

Tag Value

An integer representing one of the options of enumeration **<BACnetRestartReason>**.

device.lastRestoreTime

According to the BACnet protocol documentation:

Indicates the time at which the device's image was last restored.

Property *lastRestoreTime* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.lastRestoreTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 157

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;

- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

device.localDate

According to the BACnet protocol documentation:

Indicates the current date to the best of the device's knowledge.

Property *localDate* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.localDate

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 56

Tag Value

A Date/Time value (only the Date part).

device.localTime

According to the BACnet protocol documentation:

Indicates the current time of day to the best of the device's knowledge.

Property *localTime* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.localTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 57

Tag Value

A Date/Time value (only the Time part).

device.location

According to the BACnet protocol documentation:

Indicates the physical location of the BACnet device.

Property *location* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.location

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 58

Tag Value

A string value.

device.maxApduLengthAccepted

According to the BACnet protocol documentation:

The maximum number of octets that may be contained in a single, indivisible application layer protocol data unit.

Property *maxApduLengthAccepted* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.maxApduLengthAccepted

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 62

Tag Value

A 32-bit unsigned integer.

device.maxInfoFrames

According to the BACnet protocol documentation:

Specifies the maximum number of information frames the node may send before it must pass the token.

Property *maxInfoFrames* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.maxInfoFrames

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 63

Tag Value

A 32-bit unsigned integer.

device.maxMaster

According to the BACnet protocol documentation:

Specifies the highest possible address for master nodes, and shall be less than or equal to 127.

Property *maxMaster* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.maxMaster

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 64

Tag Value

A 32-bit unsigned integer.

device.maxSegmentsAccepted

According to the BACnet protocol documentation:

Indicates the maximum number of segments of an APDU that this device will accept.

Property *maxSegmentsAccepted* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.maxSegmentsAccepted

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 167

Tag Value

A 32-bit unsigned integer.

device.modelName

According to the BACnet protocol documentation:

Assigned by the vendor to represent the model of the BACnet device.

Property *modelName* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.modelName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 70

Tag Value

A string value.

device.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

device.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

device.numberOfApduRetries

According to the BACnet protocol documentation:

Indicates the maximum number of times that an APDU shall be retransmitted.

Property *numberOfApduRetries* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.numberOfApduRetries

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 73

Tag Value

A 32-bit unsigned integer.

device.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

device.objectList

According to the BACnet protocol documentation:

An array of object identifiers, one for each object within the device that is accessible through BACnet services.

Property *objectList* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.objectList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 76

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetObjectIdentifier>**.

device.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

device.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

device.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

device.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

device.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

device.protocolObjectTypesSupported

According to the BACnet protocol documentation:

Indicates which standardized object types are supported by this device's protocol implementation.

Property *protocolObjectTypesSupported* of object *device* can be read and written using a block tag with 60 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.protocolObjectTypesSupported
Size	60 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 8
B3:	<objectAddress>
B4:	<property> = 96
Size:	60 elements

Block Elements

Index	Name	Type
0	analogInput	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	analogOutput	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	analogValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	binaryInput	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
4	binaryOutput	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
5	binaryValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
6	calendar	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

7	command	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
8	device	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
9	eventEnrollment	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
10	file	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
11	group	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
12	loop	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
13	multiStateInput	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
14	multiStateOutput	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
15	notificationClass	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
16	program	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
17	schedule	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
18	averaging	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
19	multiStateValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
20	trendLog	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
21	lifeSafetyPoint	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
22	lifeSafetyZone	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
23	accumulator	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
24	pulseConverter	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
25	eventLog	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
26	globalGroup	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
27	trendLogMultiple	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
28	loadControl	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
29	structuredView	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
30	accessDoor	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
31	timer	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
32	accessCredential	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
33	accessPoint	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
34	accessRights	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
35	accessUser	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
36	accessZone	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
37	credentialDataInput	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

38	networkSecurity	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
39	bitstringValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
40	characterstringValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
41	datepatternValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
42	dateValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
43	datetimepatternValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
44	datetimeValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
45	integerValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
46	largeAnalogValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
47	octetstringValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
48	positiveIntegerValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
49	timepatternValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
50	timeValue	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
51	notificationForwarder	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
52	alertEnrollment	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
53	channel	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
54	lightingOutput	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
55	binaryLightingOutput	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
56	networkPort	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
57	elevatorGroup	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
58	escalator	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
59	lift	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

device.protocolRevision

According to the BACnet protocol documentation:

Indicates the minor revision level of the BACnet standard.

Property *protocolRevision* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.protocolRevision

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 139

Tag Value

A 32-bit unsigned integer.

device.protocolServicesSupported

According to the BACnet protocol documentation:

Indicates which standardized protocol services are supported by this device's protocol implementation.

Property *protocolServicesSupported* of object *device* can be read and written using a block tag with 44 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.protocolServicesSupported
Size	44 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 8
B3:	<objectAddress>
B4:	<property> = 97
Size:	44 elements

Block Elements

Index	Name	Type
0	acknowledgeAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	confirmedControlNotification	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

2	confirmedEventNotification	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	getAlarmSummary	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
4	getEnrollmentSummary	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
5	subscribeCover	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
6	atomicReadFile	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
7	atomicWriteFile	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
8	addListElement	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
9	removeListElement	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
10	createObject	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
11	deleteObject	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
12	readProperty	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
13	_unmappedBit13	This element is unused.
14	readPropertyMultiple	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
15	writeProperty	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
16	writePropertyMultiple	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
17	deviceCommunicationControl	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
18	confirmedPrivateTransfer	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
19	confirmedTextMessage	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
20	reinitializeDevice	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
21	vtOpen	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
22	vtClose	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
23	vtData	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
24	_unmappedBit24	This element is unused.
25	_unmappedBit25	This element is unused.
26	iAm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

27	iHave	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
28	unconfirmedCovNotification	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
29	unconfirmedEventNotification	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
30	unconfirmedPrivateTransfer	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
31	unconfirmedTextMessage	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
32	timeSynchronization	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
33	whoHas	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
34	whols	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
35	readRange	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
36	utcTimeSynchronization	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
37	lifeSafetyOperation	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
38	subscribeCovProperty	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
39	getEventInformation	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
40	writeGroup	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
41	subscribeCovPropertyMultiple	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
42	confirmedCovNotificationMultiple	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
43	unconfirmedCovNotificationMultiple	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

device.protocolVersion

According to the BACnet protocol documentation:

Represents the version of the BACnet protocol supported by this BACnet device.

Property *protocolVersion* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.protocolVersion

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 98

Tag Value

A 32-bit unsigned integer.

device.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

device.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

device.restartNotificationRecipients

According to the BACnet protocol documentation:

Used to control the restrictions on which devices, if any, are to be notified when a restart occurs. The value of this property shall be a list of zero or more BACnetRecipients. If the list is of length zero, a device is prohibited from sending a device restart notification. The default value of the property shall be a single entry representing a broadcast on the local network.

Property *restartNotificationRecipients* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.restartNotificationRecipients

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 202

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetRecipient>**.

device.restoreCompletionTime

According to the BACnet protocol documentation:

Indicates the amount of time in seconds that the device is allowed to remain unresponsive after the sending of a ReinitializeDevice-ACK at the end of a restore procedure. The restoring device shall either wait or be prepared to encounter communication timeouts during this period.

Property `restoreCompletionTime` of object `device` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.restoreCompletionTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 340

Tag Value

A 16-bit integer value ranging from 0 to 65535.

device.restorePreparationTime

According to the BACnet protocol documentation:

Indicates the amount of time in seconds that the device is allowed to remain unresponsive after the sending of a ReinitializeDevice-ACK at the start of a restore procedure. The restoring device shall either wait or be prepared to encounter communication timeouts during this period.

Property `restorePreparationTime` of object `device` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.restorePreparationTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 341

Tag Value

A 16-bit integer value ranging from 0 to 65535.

device.segmentationSupported

According to the BACnet protocol documentation:

Indicates whether the BACnet Device supports segmentation of messages, and, if so, whether it supports segmented transmission, reception, or both. 0 = both, 1 = transmit, 2 = receive, 3 = no segmentation

Property *segmentationSupported* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.segmentationSupported

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 107

Tag Value

An integer representing one of the options of enumeration **<BACnetSegmentation>**.

device.serialNumber

According to the BACnet protocol documentation:

Assigned by the vendor to represent the serial number in a vendor-specific model series. The combination of Model_Name, Vendor_Identifier and Serial_Number uniquely identifies a device.

Property *serialNumber* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.serialNumber

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 372

Tag Value

A string value.

device.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *device* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 8
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

device.structuredObjectList

According to the BACnet protocol documentation:

Contains references to objects chosen for use as starting points for the traversal of object hierarchies.

Property *structuredObjectList* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.structuredObjectList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 209

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetObjectIdentifier>**.

device.systemStatus

According to the BACnet protocol documentation:

Reflects the current physical and logical status of the BACnet Device. 0 = operational, 1 = operational (read-only), 2 = download required, 3 = download in progress, 4 = non operational

Property *systemStatus* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.systemStatus

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 112

Tag Value

An integer representing one of the options of enumeration **<BACnetDeviceStatus>**.

device.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

device.timeOfDeviceRestart

According to the BACnet protocol documentation:

Contains the time at which the device was last restarted.

Property *timeOfDeviceRestart* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.timeOfDeviceRestart

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 203

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

device.timeSynchronizationInterval

According to the BACnet protocol documentation:

Specifies the periodic interval in minutes at which TimeSynchronization and UTCTimeSynchronization requests shall be sent.

Property *timeSynchronizationInterval* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.timeSynchronizationInterval

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 204

Tag Value

A 32-bit unsigned integer.

device.timeSynchronizationRecipients

According to the BACnet protocol documentation:

Used to control the restrictions placed on a device's use of the TimeSynchronization service.

Property *timeSynchronizationRecipients* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.timeSynchronizationRecipients

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 116

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetRecipient>**.

device.utcOffset

According to the BACnet protocol documentation:

Indicates the number of minutes (-720 to +720) offset between local standard time and Universal Time Coordinated. The time zones to the west of the zero degree meridian shall be positive values, and those to the east shall be negative values.

Property *utcOffset* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.utcOffset

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 119

Tag Value

A 32-bit signed integer.

device.utcTimeSynchronizationRecipients

According to the BACnet protocol documentation:

Controls the restrictions placed on a device's use of the UTCTimeSynchronization service. The value of this property shall be a list of zero or more BACnetRecipients. If the list is of length zero, or the property is not present, the device is prohibited from automatically sending a UTCTimeSynchronization request. If the list is of length one or more, the device may automatically send a UTCTimeSynchronization request but only to the devices or addresses listed.

Property *utcTimeSynchronizationRecipients* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.utcTimeSynchronizationRecipients

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 206

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetRecipient>**.

device.vendorIdentifier

According to the BACnet protocol documentation:

A unique vendor identification code, assigned by ASHRAE, which is used to distinguish proprietary extensions of the protocol.

Property *vendorIdentifier* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.vendorIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 120

Tag Value

A 16-bit integer value ranging from 0 to 65535.

device.vendorName

According to the BACnet protocol documentation:

Identifies the manufacturer of the BACnet Device.

Property *vendorName* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.vendorName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 121

Tag Value

A string value.

device.vtClassesSupported

According to the BACnet protocol documentation:

List of enumerations, each of them indicating a particular set of terminal characteristics.

Property *vtClassesSupported* of object *device* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	device-<objectAddress>.vtClassesSupported

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 8
N3:	<objectAddress>
N4:	<property> = 122

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetVTClass>**.

elevatorGroup Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a group of lifts or escalators (a group being defined as those lifts or escalators controlled by a single supervisory controller).

Objects of type *elevatorGroup* have the following properties:

- **description**
- **groupId**
- **groupMembers**
- **groupMode**
- **landingCallControl**
- **landingCalls**
- **machineRoomId**
- **objectIdentifier**
- **objectName**
- **objectType**
- **profileLocation**

- **profileName**
- **propertyList**
- **tags**

elevatorGroup.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *elevatorGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 57
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

elevatorGroup.groupId

According to the BACnet protocol documentation:

Represents the identification number for the group of lifts or escalators represented by this object. This identification number shall be unique for the groups in this machine room, but might not be otherwise unique in the building.

Property *groupId* of object *elevatorGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.groupId

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 57
N3:	<objectAddress>
N4:	<property> = 465

Tag Value

An 8-bit integer value ranging from 0 to 255.

elevatorGroup.groupMembers

According to the BACnet protocol documentation:

Defines the members of the group. If the optional device identifier is not present for a particular group member, then that object shall reside in the same device that maintains the Global Group object.

Property *groupMembers* of object *elevatorGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.groupMembers

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 57
N3:	<objectAddress>
N4:	<property> = 345

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetObjectIdentifier>**.

elevatorGroup.groupMode

According to the BACnet protocol documentation:

Conveys the operating mode of the group of lifts. This is used to represent some special traffic modes of control of the supervisory controller of a group of lifts. Supervisory controllers are not required to support all modes. Under a special traffic mode, the car dispatching algorithm may be different.

Property *groupMode* of object *elevatorGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.groupMode

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 57
N3:	<objectAddress>
N4:	<property> = 467

Tag Value

An integer representing one of the options of enumeration **<BACnetLiftGroupMode>**.

elevatorGroup.landingCallControl

According to the BACnet protocol documentation:

A write to this property is equivalent to a passenger pressing a call button at a landing, indicating either desired direction of travel or destination floor.

Property *landingCallControl* of object *elevatorGroup* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.landingCallControl
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 57
B3:	<objectAddress>
B4:	<property> = 471
Size:	3 elements

Block Elements

Index	Name	Type
0	FloorNumber	An 8-bit integer value ranging from 0 to 255.
1	Command	One of the following data types: - An integer representing one of the options of enumeration <BACnetLiftCarDirection>. - A character string containing an XML with a value of type <Unsigned8>.
2	FloorText	A string value. OBS: this is an optional element, a <null> value indicates the element is not present.

elevatorGroup.landingCalls

According to the BACnet protocol documentation:

Each element of this list shall represent a currently active call for the group of lifts.

Property *landingCalls* of object *elevatorGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.landingCalls

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 57
N3:	<objectAddress>
N4:	<property> = 470

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLandingCallStatus>**.

elevatorGroup.machineRoomId

According to the BACnet protocol documentation:

References the Positive Integer Value Object whose Present_Value property contains the identification number for the machine room that contains the group of lifts or escalators represented by this object. If there is no such identification number, this property shall contain an object instance number of 4194303.

Property *machineRoomId* of object *elevatorGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.machineRoomId

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 57
N3:	<objectAddress>
N4:	<property> = 474

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

elevatorGroup.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *elevatorGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 57
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

elevatorGroup.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *elevatorGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 57
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

elevatorGroup.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *elevatorGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 57
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

elevatorGroup.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *elevatorGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 57
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

elevatorGroup.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *elevatorGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 57
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

elevatorGroup.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *elevatorGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 57
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

elevatorGroup.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *elevatorGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	elevatorGroup-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 57
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

escalator Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of an escalator.

Objects of type escalator have the following properties:

- **ackedTransitions**
- **description**
- **elevatorGroup**
- **energyMeter**
- **energyMeterRef**
- **escalatorMode**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultSignals**
- **groupId**
- **installationId**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **operationDirection**
- **outOfService**
- **passengerAlarm**
- **powerMode**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**

- **tags**
- **timeDelay**
- **timeDelayNormal**

escalator.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *escalator* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 58
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

escalator.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

escalator.elevatorGroup

According to the BACnet protocol documentation:

References the Elevator Group object whose Group_Members property contains a reference to this Lift object. If there is no such Elevator Group object, this property shall contain an object instance of 4194303.

Property *elevatorGroup* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.elevatorGroup

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 459

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

escalator.energyMeter

According to the BACnet protocol documentation:

Indicates the accumulated energy consumption by the lift. The units shall be kilowatt-hours. When this value reaches 99999 kWh, it shall wrap to a value near zero; the particular value to which it wraps is a local matter. If the Energy_Meter_Ref property is present and initialized (contains an instance other than 4194303), then the Energy_Meter property, if present, shall contain a value of 0.0.

Property *energyMeter* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.energyMeter

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 460

Tag Value

A 32-bit floating point number.

escalator.energyMeterRef

According to the BACnet protocol documentation:

References the object which indicates the accumulated energy consumption by the lift.

Property *energyMeterRef* of object *escalator* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.energyMeterRef
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 58
B3:	<objectAddress>
B4:	<property> = 461
Size:	2 elements

Block Elements

Index	Name	Type
0	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.
1	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

escalator.escalatorMode

According to the BACnet protocol documentation:

Indicates the current operational mode of the escalator.

Property *escalatorMode* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.escalatorMode

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 462

Tag Value

An integer representing one of the options of enumeration **<BACnetEscalatorMode>**.

escalator.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

escalator.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

escalator.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

escalator.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *escalator* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 58
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

escalator.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *escalator* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 58
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

escalator.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *escalator* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 58
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

escalator.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

escalator.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *escalator* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 58
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

escalator.faultSignals

According to the BACnet protocol documentation:

Represents a list of values that indicates fault conditions of the lift.

Property *faultSignals* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.faultSignals

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 463

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetEscalatorFault>**.

escalator.groupId

According to the BACnet protocol documentation:

Represents the identification number for the group of lifts or escalators represented by this object. This identification number shall be unique for the groups in this machine room, but might not be otherwise unique in the building.

Property *groupId* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.groupId

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 465

Tag Value

An 8-bit integer value ranging from 0 to 255.

escalator.installationId

According to the BACnet protocol documentation:

Represents the identification number for the lift represented by this object. This identification number shall be unique for the lift in this group, but might not be otherwise unique for other lifts in the machine room or the building.

Property *installationId* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.installationId

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 469

Tag Value

An 8-bit integer value ranging from 0 to 255.

escalator.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

escalator.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

escalator.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

escalator.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

escalator.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

escalator.operationDirection

According to the BACnet protocol documentation:

Represents the direction and speed in which this escalator is presently moving.

Property *operationDirection* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.operationDirection

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 477

Tag Value

An integer representing one of the options of enumeration **<BACnetEscalatorOperationDirection>**.

escalator.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

escalator.passengerAlarm

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the passenger alarm has been activated.

Property *passengerAlarm* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.passengerAlarm

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 478

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

escalator.powerMode

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the escalator is powered (independent of whether it is moving).

Property *powerMode* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.powerMode

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 479

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

escalator.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

escalator.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

escalator.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

escalator.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

escalator.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

escalator.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *escalator* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 58
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

escalator.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

escalator.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

escalator.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *escalator* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	escalator-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 58
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

eventEnrollment Object

According to the BACnet protocol documentation:

Defines a standardized object that represents and contains the information required for algorithmic reporting of events.

Objects of type eventEnrollment have the following properties:

- **ackedTransitions**
- **description**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventParameters**
- **eventState**
- **eventTimeStamps**
- **eventType**
- **faultParameters**
- **faultType**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectPropertyReference**
- **objectType**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**
- **tags**

- **timeDelayNormal**

eventEnrollment.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *eventEnrollment* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 9
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventEnrollment.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

eventEnrollment.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `eventEnrollment` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventEnrollment.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property `eventAlgorithmInhibitRef` of object `eventEnrollment` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

eventEnrollment.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventEnrollment.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property `eventEnable` of object `eventEnrollment` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 9
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventEnrollment.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property `eventMessageTexts` of object `eventEnrollment` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 9
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

eventEnrollment.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *eventEnrollment* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 9
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

eventEnrollment.eventParameters

According to the BACnet protocol documentation:

Provides the parameter values needed for the algorithm configured in eventType.

Property *eventParameters* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.eventParameters

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 83

Tag Value

One of the following data types:

- A character string containing an XML with a value of type **<BACnetEventParameter_ChangeOfBitstring>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_ChangeOfState>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_ChangeOfValue>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_CommandFailure>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_FloatingLimit>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_OutOfRange>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_ChangeOfLifeSafety>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_Extended>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_BufferReady>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_UnsignedRange>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_AccessEvent>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_DoubleOutOfRange>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_SignedOutOfRange>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_UnsignedOutOfRange>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_ChangeOfCharacterstring>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_ChangeOfStatusFlags>**.

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetEventParameter_ChangeOfDiscreteValue>**.
- A character string containing an XML with a value of type **<BACnetEventParameter_ChangeOfTimer>**.

eventEnrollment.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

eventEnrollment.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *eventEnrollment* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 9
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

eventEnrollment.eventType

According to the BACnet protocol documentation:

Indicates the type of event algorithm that is to be used to detect the occurrence of events and report to enrolled devices. 0 = change of bit string, 1 = change of state, 2 = change of value, 3 = command failure, 4 = floating limit, 5 = out of range

Property *eventType* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.eventType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 37

Tag Value

An integer representing one of the options of enumeration **<BACnetEventType>**.

eventEnrollment.faultParameters

According to the BACnet protocol documentation:

Determines the fault algorithm used to monitor the referenced object and provides the parameter values needed for this fault algorithm. If the Event Enrollment object does not apply a fault algorithm, then the fault parameter choice NONE shall be set in this property.

Property *faultParameters* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.faultParameters

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 358

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetFaultParameter_FaultCharacterstring>**.
- A character string containing an XML with a value of type **<BACnetFaultParameter_FaultExtended>**.
- A character string containing an XML with a value of type **<BACnetFaultParameter_FaultLifeSafety>**.
- A character string containing an XML with a value of type **<BACnetFaultParameter_FaultState>**.
- A character string containing an XML with a value of type **<BACnetFaultParameter_FaultStatusFlags>**.
- A character string containing an XML with a value of type **<BACnetFaultParameter_FaultOutOfRange>**.
- A character string containing an XML with a value of type **<BACnetFaultParameter_FaultListed>**.

eventEnrollment.faultType

According to the BACnet protocol documentation:

Indicates the type of fault algorithm that is applied by the object.

Property *faultType* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.faultType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 359

Tag Value

An integer representing one of the options of enumeration **<BACnetFaultType>**.

eventEnrollment.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

eventEnrollment.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

eventEnrollment.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

eventEnrollment.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

eventEnrollment.objectPropertyReference

According to the BACnet protocol documentation:

Designates the particular object and property referenced by this object.

Property *objectPropertyReference* of object *eventEnrollment* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.objectPropertyReference
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 9
B3:	<objectAddress>
B4:	<property> = 78
Size:	4 elements

Block Elements

Index	Name	Type
0	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').
1	PropertyIdentifier	An integer representing one of the options of enumeration <BACnetPropertyIdentifier> .
2	PropertyArrayIndex	A 32-bit unsigned integer. OBS: this is an optional element, a <null> value indicates the element is not present.
3	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.

eventEnrollment.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

eventEnrollment.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

eventEnrollment.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

eventEnrollment.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

eventEnrollment.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

eventEnrollment.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall

have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventEnrollment.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *eventEnrollment* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 9
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventEnrollment.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

eventEnrollment.timeDelayNormal

According to the BACnet protocol documentation:

*Contains the *pTimeDelayNormal* parameter for the object's event algorithm.*

Property *timeDelayNormal* of object *eventEnrollment* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventEnrollment-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 9
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

eventLog Object

According to the BACnet protocol documentation:

Records event notifications with timestamps and other pertinent data in an internal buffer for subsequent retrieval. Each timestamped buffer entry is called a "log record."

Objects of type eventLog have the following properties:

- **ackedTransitions**
- **bufferSize**
- **description**
- **enable**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **lastNotifyRecord**
- **logBuffer**
- **notificationClass**
- **notificationThreshold**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**

- **profileLocation**
- **profileName**
- **propertyList**
- **recordCount**
- **recordsSinceNotification**
- **reliability**
- **reliabilityEvaluationInhibit**
- **startTime**
- **statusFlags**
- **stopTime**
- **stopWhenFull**
- **tags**
- **totalRecordCount**

eventLog.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *eventLog* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 25
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventLog.bufferSize

According to the BACnet protocol documentation:

Specifies the maximum number of log records the log buffer may hold. If writable, it may not be written when Enable is TRUE. The disposition of existing log records when Buffer_Size is written is a local matter. If all records are deleted when the Buffer_Size is written then the object shall act as if the Record_Count was set to zero.

Property *bufferSize* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.bufferSize

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 126

Tag Value

A 32-bit unsigned integer.

eventLog.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

eventLog.enable

According to the BACnet protocol documentation:

Indicates and controls whether (TRUE) or not (FALSE) logging of events is enabled. Logging occurs if and only if Enable is TRUE, Local_Time is on or after Start_Time, and Local_Time is before Stop_Time. If Start_Time contains an unspecified datetime, then it shall be considered equal to 'the start of time'. If Stop_Time contains an unspecified datetime, then it shall be considered equal to 'the end of time'. Log records of type log-status are recorded without regard to the value of the Enable property.

Property *enable* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.enable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 133

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventLog.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventLog.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

eventLog.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventLog.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *eventLog* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 25
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventLog.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *eventLog* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 25
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

eventLog.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *eventLog* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 25
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

eventLog.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

eventLog.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *eventLog* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 25
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

eventLog.lastNotifyRecord

According to the BACnet protocol documentation:

Represents the pPreviousCount parameter of the object's event algorithm.

Property *lastNotifyRecord* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.lastNotifyRecord

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 173

Tag Value

A 32-bit unsigned integer.

eventLog.logBuffer

According to the BACnet protocol documentation:

Contains a list of up to Buffer_Size timestamped log records of datatype BACnetEventLogRecord, each of which conveys the event notification parameters or status changes in the Event Log object.

Property *logBuffer* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.logBuffer

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 131

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetEventLogRecord>**.

eventLog.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

eventLog.notificationThreshold

According to the BACnet protocol documentation:

Contains the pThreshold parameter for the object's event algorithm.

Property *notificationThreshold* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.notificationThreshold

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 137

Tag Value

A 32-bit unsigned integer.

eventLog.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

eventLog.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

eventLog.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

eventLog.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

eventLog.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

eventLog.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *fileName* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.fileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

eventLog.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

eventLog.recordCount

According to the BACnet protocol documentation:

Represents the number of records currently resident in the log buffer.

Property *recordCount* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.recordCount

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 141

Tag Value

A 32-bit unsigned integer.

eventLog.recordsSinceNotification

According to the BACnet protocol documentation:

Represents the number of log records collected since the previous notification, or since the beginning of logging if no previous notification has occurred.

Property *recordsSinceNotification* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.recordsSinceNotification

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 140

Tag Value

A 32-bit unsigned integer.

eventLog.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

eventLog.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventLog.startTime

According to the BACnet protocol documentation:

Specifies the date and time at or after which logging shall be enabled by this property. If this property contains an unspecified datetime, then the conditions for logging to be enabled by Start_Time shall be ignored. If Start_Time specifies a date and time after Stop_Time, then logging shall be disabled. This property shall be writable if present. When Start_Time is reached, the value of the Enable property is not changed.

Property *startTime* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.startTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 142

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

eventLog.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *eventLog* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 25
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventLog.stopTime

According to the BACnet protocol documentation:

Specifies the date and time at or after which logging shall be disabled by this property. If this property contains an unspecified datetime, then the conditions for logging to be disabled by Stop_Time shall be ignored. If Stop_Time specifies a date and time earlier than Start_Time, then logging shall be disabled. This property shall be writable if present. When Stop_Time is reached, the value of the Enable property is not changed.

Property *stopTime* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.stopTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 143

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

eventLog.stopWhenFull

According to the BACnet protocol documentation:

Specifies whether (TRUE) or not (FALSE) logging should cease when the log buffer is full. When logging ceases because the addition of one more log record would cause the log buffer to be full, Enable shall be set to FALSE and the event recorded. If Stop_When_Full is writable, attempts to write the value TRUE to the Stop_When_Full property while Record_Count is equal to Buffer_Size shall result in the oldest log record in the log buffer being discarded, and shall cause the Enable property to be set to FALSE and the event to be recorded.

Property *stopWhenFull* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.stopWhenFull

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 144

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

eventLog.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

eventLog.totalRecordCount

According to the BACnet protocol documentation:

Represents the total number of log records collected by the Trend Log object since creation. When the value of Total_Record_Count reaches its maximum possible value of 232 - 1, the next value it takes shall be one. Once this value has wrapped to one, its semantic value (the total number of log records collected) has been lost but its use in generating notifications remains.

Property *totalRecordCount* of object *eventLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	eventLog-<objectAddress>.totalRecordCount

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 25
N3:	<objectAddress>
N4:	<property> = 145

Tag Value

A 32-bit unsigned integer.

file Object

According to the BACnet protocol documentation:

Defines a standardized object that is used to describe properties of data files that may be accessed using File Services.

Objects of type file have the following properties:

- **archive**
- **description**
- **fileAccessMethod**
- **fileSize**
- **fileType**
- **modificationDate**
- **objectIdentifier**
- **objectName**
- **objectType**
- **profileLocation**
- **profileName**
- **propertyList**
- **readOnly**
- **recordCount**
- **tags**

file.archive

According to the BACnet protocol documentation:

Indicates whether the File object has been saved for historical or backup purposes. This property shall be logical TRUE only if no changes have been made to the file data by internal processes or through File Access Services since the last time the object was archived.

Property *archive* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.archive

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 13

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

file.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

file.fileAccessMethod

According to the BACnet protocol documentation:

Indicates the type(s) of file access supported for this object. 0 = record access, 1 = stream access, 2 = record and stream access

Property *fileAccessMethod* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.fileAccessMethod

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 41

Tag Value

An integer representing one of the options of enumeration **<BACnetFileAccessMethod>**.

file.fileSize

According to the BACnet protocol documentation:

Indicates the size of the file data in octets.

Property *fileSize* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.fileSize

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 42

Tag Value

A 32-bit unsigned integer.

file.fileType

According to the BACnet protocol documentation:

String that identifies the intended use of this file.

Property *fileType* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.fileType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 43

Tag Value

A string value.

file.modificationDate

According to the BACnet protocol documentation:

Indicates the last time this object was modified.

Property *modificationDate* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.modificationDate

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 71

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

file.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

file.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

file.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

file.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

file.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *fileName* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.fileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

file.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

file.readOnly

According to the BACnet protocol documentation:

Indicates whether (FALSE) or not (TRUE) the file data may be changed through the use of a BACnet AtomicWriteFile service.

Property *readOnly* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.readOnly

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 99

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

file.recordCount

According to the BACnet protocol documentation:

Represents the number of records currently resident in the log buffer.

Property *recordCount* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.recordCount

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 141

Tag Value

A 32-bit unsigned integer.

file.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *file* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	file-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 10
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

globalGroup Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent a collection of other objects and one or more of their properties. A Global Group object is used to simplify the exchange of information between BACnet devices by providing a shorthand way to specify all members of the group at once. A Global Group object differs from a Group object in that its members can be from anywhere in the BACnet internetwork, it supports intrinsic event reporting, and it exposes a method for sending periodic COV notifications. The Global Group object is able to monitor all referenced Status_Flags properties to detect changes to non-normal states and can initiate an event notification message conveying the values of all of the members of the group. This provides a mechanism to define a large set of property values that are made available when an event occurs.

Objects of type globalGroup have the following properties:

- **ackedTransitions**
- **clientCovIncrement**
- **covResubscriptionInterval**
- **covuPeriod**
- **covuRecipients**
- **description**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**

- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **groupMemberNames**
- **groupMembers**
- **memberStatusFlags**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **requestedUpdateInterval**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **updateInterval**

globalGroup.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *globalGroup* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 26
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

globalGroup.clientCovIncrement

According to the BACnet protocol documentation:

Specifies the increment to be used in determining that a change of value has occurred.

Property *clientCovIncrement* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.clientCovIncrement

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 127

Tag Value

One of the following data types:

- A 32-bit floating point number.

- A null/empty value.

globalGroup.covResubscriptionInterval

According to the BACnet protocol documentation:

Specifies the number of seconds between COV resubscriptions, provided that COV subscription is in effect.

Property *covResubscriptionInterval* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.covResubscriptionInterval

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 128

Tag Value

A 32-bit unsigned integer.

globalGroup.covuPeriod

According to the BACnet protocol documentation:

Indicates the amount of time in seconds between the periodic unsubscribed COV notifications performed by this object. These COV notifications convey the value of the Present_Value and Member_Status_Flags properties. If the value of COVU_Period is zero, then periodic unsubscribed COV notification messages shall not be transmitted.

Property *covuPeriod* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.covuPeriod

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 349

Tag Value

A 32-bit unsigned integer.

globalGroup.covRecipients

According to the BACnet protocol documentation:

Used to control the restrictions on which devices, if any, are to receive periodic unsubscribed COV notifications for the Present_Value and Member_Status_Flags properties. This property is required if the object sends such notifications. The value of this property shall be a list of zero or more BACnetRecipients. If the list is of length zero, a device is prohibited from sending such notifications. If the list is of length one or more, the device shall send the notifications, but only to the devices or addresses listed.

Property *covRecipients* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.covRecipients

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 350

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetRecipient>**.

globalGroup.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

globalGroup.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `globalGroup` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

globalGroup.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property `eventAlgorithmInhibitRef` of object `globalGroup` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

globalGroup.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

globalGroup.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *globalGroup* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 26
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

globalGroup.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *globalGroup* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 26
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

globalGroup.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *globalGroup* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 26
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

globalGroup.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

globalGroup.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *globalGroup* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 26
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

globalGroup.groupMemberNames

According to the BACnet protocol documentation:

Represents a descriptive name for the members of the Global Group. The number of names matches the number of members defined in Group_Members. The array index of the name shall match the array index of the corresponding group member.

Property *groupMemberNames* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.groupMemberNames

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 346

Tag Value

A character string containing an XML with a value of type **<SequenceOfCharacterString>**.

globalGroup.groupMembers

According to the BACnet protocol documentation:

Defines the members of the group. If the optional device identifier is not present for a particular group member, then that object shall reside in the same device that maintains the Global Group object.

Property *groupMembers* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.groupMembers

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 345

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectPropertyReference>**.

globalGroup.memberStatusFlags

According to the BACnet protocol documentation:

Contains a logical combination of all the Status_Flags properties contained in the Present_Value. The logical combination means that each of the flags in this property (IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE) is TRUE if and only if the corresponding flag is set in any of the Status_Flags property values in the Present_Value property. This property shall be updated whenever new Status_Flags property values are updated in the Present_Value.

Property *memberStatusFlags* of object *globalGroup* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.memberStatusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 26
B3:	<objectAddress>
B4:	<property> = 347
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

globalGroup.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

globalGroup.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

globalGroup.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

globalGroup.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

globalGroup.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

globalGroup.outOfService

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

globalGroup.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyAccessResult>**.

globalGroup.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

globalGroup.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

globalGroup.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

globalGroup.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

globalGroup.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

globalGroup.requestedUpdateInterval

According to the BACnet protocol documentation:

Indicates the requested period of time between updates to Present_Value, measured in hundredths of a second when the object is not out-of-service.

Property *requestedUpdateInterval* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.requestedUpdateInterval

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 348

Tag Value

A 32-bit unsigned integer.

globalGroup.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *globalGroup* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 26
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

globalGroup.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

globalGroup.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *globalGroup* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

globalGroup.timeDelayNormal

According to the BACnet protocol documentation:

Contains the `pTimeDelayNormal` parameter for the object's event algorithm.

Property `timeDelayNormal` of object `globalGroup` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

globalGroup.updateInterval

According to the BACnet protocol documentation:

Indicates the maximum period of time between updates to the `presentValue` in hundredths of a second when the input is not overridden and not out-of-service.

Property `updateInterval` of object `globalGroup` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	globalGroup-<objectAddress>.updateInterval

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 26
N3:	<objectAddress>
N4:	<property> = 118

Tag Value

A 32-bit unsigned integer.

group Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent a collection of other objects and one or more of their properties. A group object is used to simplify the exchange of information between BACnet devices by providing a shorthand way to specify all members of the group at once. A group may be formed using any combination of object types.

Objects of type group have the following properties:

- **description**
- **listOfGroupMembers**
- **objectIdentifier**
- **objectName**
- **objectType**
- **presentValue**
- **profileLocation**
- **profileName**
- **propertyList**
- **tags**

group.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *group* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	group-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 11
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

group.listOfGroupMembers

According to the BACnet protocol documentation:

List of one or more read access specifications, which defines the members of the group that shall be referenced when this object is specified in a protocol transaction.

Property *listOfGroupMembers* of object *group* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	group-<objectAddress>.listOfGroupMembers

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 11
N3:	<objectAddress>
N4:	<property> = 53

Tag Value

A character string containing an XML with a value of type **<SequenceOfReadAccessSpecification>**.

group.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *group* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	group-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 11
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

group.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *group* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	group-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 11
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

group.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *group* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	group-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 11
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

group.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *group* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	group-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 11
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A character string containing an XML with a value of type **<SequenceOfReadAccessResult>**.

group.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *group* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	group-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 11
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

group.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *group* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	group-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 11
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

group.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *group* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	group-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 11
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

group.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *group* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	group-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 11
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

integerValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use an Integer Value object to make any kind of signed integer data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

Objects of type integerValue have the following properties:

- **ackedTransitions**
- **commandTimeArray**
- **covIncrement**
- **currentCommandPriority**
- **deadband**
- **description**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultHighLimit**
- **faultLowLimit**
- **highLimit**
- **lastCommandTime**
- **limitEnable**
- **lowLimit**
- **maxPresValue**
- **minPresValue**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityArray**
- **profileLocation**

- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **resolution**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **units**
- **valueSource**
- **valueSourceArray**

integerValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *integerValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 45
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

integerValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *integerValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 45
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

		• A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
7	elm07	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
8	elm08	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

		• A character string containing an XML with a value of type <BACnetTimeStamp>.
9	elm09	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
10	elm10	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
11	elm11	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
12	elm12	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
13	elm13	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
14	elm14	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
15	elm15	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set;

- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

integerValue.covIncrement

According to the BACnet protocol documentation:

Specifies the minimum change in presentValue that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

Property *covIncrement* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.covIncrement

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 22

Tag Value

A 32-bit unsigned integer.

integerValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

integerValue.deadband

According to the BACnet protocol documentation:

Specifies a range between the highLimit and lowLimit properties, which the presentValue must remain within for a TO-NORMAL event to be generated.

Property *deadband* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.deadband

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 25

Tag Value

A 32-bit unsigned integer.

integerValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

integerValue.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `integerValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

integerValue.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property `Event_Algorithm_Inhibit`. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type `BACnetBinaryPV` or `BOOLEAN`.

Property `eventAlgorithmInhibitRef` of object `integerValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

integerValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, `Event_State` shall be `NORMAL`, and the properties `Acked_Transitions`, `Event_Time_Stamps`, and `Event_Message_Texts` shall be equal to their respective initial conditions.

Property `eventDetectionEnable` of object `integerValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

integerValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *integerValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 45
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

integerValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *integerValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 45
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

integerValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *integerValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 45
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

integerValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

integerValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *integerValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 45
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

integerValue.faultHighLimit

According to the BACnet protocol documentation:

Specifies a limit that the Present_Value must exceed before a fault event is generated by the FAULT_OUT_OF_RANGE fault algorithm.

Property *faultHighLimit* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.faultHighLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 388

Tag Value

A 32-bit signed integer.

integerValue.faultLowLimit

According to the BACnet protocol documentation:

Specifies a limit that the Present_Value must fall below before a fault event is generated by the FAULT_OUT_OF_RANGE fault algorithm.

Property *faultLowLimit* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.faultLowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 389

Tag Value

A 32-bit signed integer.

integerValue.highLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must exceed before an event is generated.

Property *highLimit* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.highLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 45

Tag Value

A 32-bit signed integer.

integerValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

integerValue.limitEnable

According to the BACnet protocol documentation:

Conveys two flags that separately enable and disable reporting of highLimit and lowLimit offnormal events and their return to normal.

Property *limitEnable* of object *integerValue* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.limitEnable
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 45
B3:	<objectAddress>
B4:	<property> = 52
Size:	2 elements

Block Elements

Index	Name	Type
0	lowLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	highLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

integerValue.lowLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must fall below before an event is generated.

Property *lowLimit* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.lowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 59

Tag Value

A 32-bit signed integer.

integerValue.maxPresValue

According to the BACnet protocol documentation:

Indicates the highest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *maxPresValue* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.maxPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 65

Tag Value

A 32-bit signed integer.

integerValue.minPresValue

According to the BACnet protocol documentation:

Indicates the lowest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *minPresValue* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.minPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 69

Tag Value

A 32-bit signed integer.

integerValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

integerValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

integerValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '**<Type>-<Address>**' (ex: 'analogOutput-345').

integerValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

integerValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

integerValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

integerValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit signed integer.

integerValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *integerValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 45
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
1	elm01	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
2	elm02	One of the following data types:

integerValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

integerValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

integerValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

integerValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

integerValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

integerValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A 32-bit signed integer.

integerValue.resolution

According to the BACnet protocol documentation:

Indicates the smallest recognizable change in presentValue in engineering units (read-only).

Property *resolution* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.resolution

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 106

Tag Value

A 32-bit signed integer.

integerValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *integerValue* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 45
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

integerValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

integerValue.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

integerValue.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

integerValue.units

According to the BACnet protocol documentation:

Indicates the measurement units of this object.

Property *units* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.units

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 117

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

integerValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *integerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 45
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

integerValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *integerValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	integerValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 45
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

largeAnalogValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a Large Analog Value object to make any kind of double-precision data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

Objects of type largeAnalogValue have the following properties:

- **ackedTransitions**
- **commandTimeArray**
- **covIncrement**
- **currentCommandPriority**
- **deadband**
- **description**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultHighLimit**
- **faultLowLimit**
- **highLimit**
- **lastCommandTime**
- **limitEnable**
- **lowLimit**
- **maxPresValue**
- **minPresValue**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**

- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **resolution**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **units**
- **valueSource**
- **valueSourceArray**

largeAnalogValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *largeAnalogValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 46
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

largeAnalogValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *largeAnalogValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 46
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

		• A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
7	elm07	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
8	elm08	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

		• A character string containing an XML with a value of type <BACnetTimeStamp>.
9	elm09	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
10	elm10	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
11	elm11	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
12	elm12	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
13	elm13	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
14	elm14	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
15	elm15	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set;

- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

largeAnalogValue.covIncrement

According to the BACnet protocol documentation:

Specifies the minimum change in presentValue that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

Property *covIncrement* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.covIncrement

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 22

Tag Value

A 64-bit floating point number.

largeAnalogValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

largeAnalogValue.deadband

According to the BACnet protocol documentation:

Specifies a range between the highLimit and lowLimit properties, which the presentValue must remain within for a TO-NORMAL event to be generated.

Property *deadband* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.deadband

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 25

Tag Value

A 64-bit floating point number.

largeAnalogValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

largeAnalogValue.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `largeAnalogValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

largeAnalogValue.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property `Event_Algorithm_Inhibit`. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type `BACnetBinaryPV` or `BOOLEAN`.

Property `eventAlgorithmInhibitRef` of object `largeAnalogValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '<index>)' can be used on array properties.

largeAnalogValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, `Event_State` shall be `NORMAL`, and the properties `Acked_Transitions`, `Event_Time_Stamps`, and `Event_Message_Texts` shall be equal to their respective initial conditions.

Property `eventDetectionEnable` of object `largeAnalogValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

largeAnalogValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *largeAnalogValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 46
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

largeAnalogValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *largeAnalogValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 46
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

largeAnalogValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *largeAnalogValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 46
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

largeAnalogValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

largeAnalogValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *largeAnalogValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 46
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

largeAnalogValue.faultHighLimit

According to the BACnet protocol documentation:

Specifies a limit that the Present_Value must exceed before a fault event is generated by the FAULT_OUT_OF_RANGE fault algorithm.

Property *faultHighLimit* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.faultHighLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 388

Tag Value

A 64-bit floating point number.

largeAnalogValue.faultLowLimit

According to the BACnet protocol documentation:

Specifies a limit that the Present_Value must fall below before a fault event is generated by the FAULT_OUT_OF_RANGE fault algorithm.

Property *faultLowLimit* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.faultLowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 389

Tag Value

A 64-bit floating point number.

largeAnalogValue.highLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must exceed before an event is generated.

Property *highLimit* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.highLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 45

Tag Value

A 64-bit floating point number.

largeAnalogValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

largeAnalogValue.limitEnable

According to the BACnet protocol documentation:

Conveys two flags that separately enable and disable reporting of highLimit and lowLimit offnormal events and their return to normal.

Property *limitEnable* of object *largeAnalogValue* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.limitEnable
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 46
B3:	<objectAddress>
B4:	<property> = 52
Size:	2 elements

Block Elements

Index	Name	Type
0	lowLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	highLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

largeAnalogValue.lowLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must fall below before an event is generated.

Property *lowLimit* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.lowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 59

Tag Value

A 64-bit floating point number.

largeAnalogValue.maxPresValue

According to the BACnet protocol documentation:

Indicates the highest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *maxPresValue* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.maxPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 65

Tag Value

A 64-bit floating point number.

largeAnalogValue.minPresValue

According to the BACnet protocol documentation:

Indicates the lowest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *minPresValue* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.minPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 69

Tag Value

A 64-bit floating point number.

largeAnalogValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

largeAnalogValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

largeAnalogValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '**<Type>-<Address>**' (ex: 'analogOutput-345').

largeAnalogValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

largeAnalogValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

largeAnalogValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

largeAnalogValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 64-bit floating point number.

largeAnalogValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *largeAnalogValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 46
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
1	elm01	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
2	elm02	One of the following data types:

largeAnalogValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

largeAnalogValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

largeAnalogValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

largeAnalogValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

largeAnalogValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

largeAnalogValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A 64-bit floating point number.

largeAnalogValue.resolution

According to the BACnet protocol documentation:

Indicates the smallest recognizable change in presentValue in engineering units (read-only).

Property *resolution* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.resolution

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 106

Tag Value

A 64-bit floating point number.

largeAnalogValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *largeAnalogValue* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 46
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

largeAnalogValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property tags of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

largeAnalogValue.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

largeAnalogValue.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

largeAnalogValue.units

According to the BACnet protocol documentation:

Indicates the measurement units of this object.

Property *units* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.units

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 117

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

largeAnalogValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *largeAnalogValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 46
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

largeAnalogValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *largeAnalogValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	largeAnalogValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 46
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

lifeSafetyPoint Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics associated with initiating and indicating devices in fire, life safety and security applications. The condition of a Life Safety Point object is represented by a mode and a state. Mode changes determine the object's inner logic and, consequently, influence the evaluation of the state. Typically, the operating mode would be under operator control. The state of the object represents the condition of the controller according to the logic internal to the device. The implementation of the logic applied to such controllers to determine the various possible states is a local matter. Typical applications of the Life Safety Point object include automatic fire detectors, pull stations, sirens, supervised printers, etc. Similar objects can be identified in security control panels.

Objects of type `lifeSafetyPoint` have the following properties:

- **acceptedModes**
- **ackedTransitions**
- **alarmValues**
- **description**
- **deviceType**
- **directReading**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultValues**
- **lifeSafetyAlarmValues**
- **maintenanceRequired**
- **memberOf**
- **mode**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **operationExpected**
- **outOfService**
- **presentValue**

- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **setting**
- **silenced**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **trackingValue**
- **units**
- **valueSource**

lifeSafetyPoint.acceptedModes

According to the BACnet protocol documentation:

Specifies all values the Mode property accepts when written to.

Property *acceptedModes* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.acceptedModes

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 175

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLifeSafetyMode>**.

lifeSafetyPoint.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *lifeSafetyPoint* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 21
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyPoint.alarmValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-OFFNORMAL event is generated.

Property *alarmValues* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.alarmValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 7

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLifeSafetyState>**.

lifeSafetyPoint.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

lifeSafetyPoint.deviceType

According to the BACnet protocol documentation:

String describing the physical device connected to the analog input. It will typically be used to describe the type of sensor attached to the analog input.

Property *deviceType* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.deviceType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 31

Tag Value

A string value.

lifeSafetyPoint.directReading

According to the BACnet protocol documentation:

Indicates an analog quantity that reflects the measured or calculated reading from an initiating device.

Property *directReading* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.directReading

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 156

Tag Value

A 32-bit floating point number.

lifeSafetyPoint.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyPoint.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

lifeSafetyPoint.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system

configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamped, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyPoint.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *lifeSafetyPoint* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 21
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyPoint.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *lifeSafetyPoint* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 21
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

lifeSafetyPoint.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *lifeSafetyPoint* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 21
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

lifeSafetyPoint.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

lifeSafetyPoint.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *lifeSafetyPoint* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 21
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

lifeSafetyPoint.faultValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-FAULT event is generated.

Property *faultValues* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.faultValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 39

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLifeSafetyState>**.

lifeSafetyPoint.lifeSafetyAlarmValues

According to the BACnet protocol documentation:

Specifies any states the PresentValue must equal before a TO-OFFNORMAL event is generated and event state LIFE_SAFETY_ALARM is entered.

Property *lifeSafetyAlarmValues* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.lifeSafetyAlarmValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 166

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLifeSafetyState>**.

lifeSafetyPoint.maintenanceRequired

According to the BACnet protocol documentation:

Indicates that maintenance is required for one or more of the life safety points that are members of this zone.

Property *maintenanceRequired* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.maintenanceRequired

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 158

Tag Value

An integer representing one of the options of enumeration **<BACnetMaintenance>**.

lifeSafetyPoint.memberOf

According to the BACnet protocol documentation:

Indicates those LifeSafetyZone objects of which this LifeSafetyPoint object is considered to be a zone member.

Property *memberOf* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.memberOf

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 159

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectReference>**.

lifeSafetyPoint.mode

According to the BACnet protocol documentation:

Conveys the desired operating mode for the Life Safety Point object.

Property *mode* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.mode

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 160

Tag Value

An integer representing one of the options of enumeration **<BACnetLifeSafetyMode>**.

lifeSafetyPoint.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

lifeSafetyPoint.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

lifeSafetyPoint.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

lifeSafetyPoint.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

lifeSafetyPoint.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

lifeSafetyPoint.operationExpected

According to the BACnet protocol documentation:

Specifies the next operation expected by this object to handle a specific life safety situation.

Property *operationExpected* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.operationExpected

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 161

Tag Value

An integer representing one of the options of enumeration **<BACnetLifeSafetyOperation>**.

lifeSafetyPoint.outOfService

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyPoint.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *lifeSafetyPoint* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

An integer representing one of the options of enumeration **<BACnetLifeSafetyState>**.

lifeSafetyPoint.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

lifeSafetyPoint.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

lifeSafetyPoint.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

lifeSafetyPoint.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

lifeSafetyPoint.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyPoint.setting

According to the BACnet protocol documentation:

Used to convey the desired setting of the input(s) or process used to determine the logical state of the Present_Value.

Property *setting* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.setting

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 162

Tag Value

An 8-bit integer value ranging from 0 to 255.

lifeSafetyPoint.silenced

According to the BACnet protocol documentation:

Indicates whether the most recently occurring transition for this object that has produced an audible or visual indication has been silenced by the receipt of a LifeSafetyOperation service request or a local process.

Property *silenced* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.silenced

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 163

Tag Value

An integer representing one of the options of enumeration **<BACnetSilencedState>**.

lifeSafetyPoint.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *lifeSafetyPoint* can be read and written using a block tag with 4 elements.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 21
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyPoint.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

lifeSafetyPoint.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

lifeSafetyPoint.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

lifeSafetyPoint.trackingValue

According to the BACnet protocol documentation:

Reflects the non-latched state of the Life Safety Point object.

Property *trackingValue* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.trackingValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 164

Tag Value

An integer representing one of the options of enumeration **<BACnetLifeSafetyState>**.

lifeSafetyPoint.units

According to the BACnet protocol documentation:

Indicates the measurement units of this object.

Property *units* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.units

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 117

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

lifeSafetyPoint.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *lifeSafetyPoint* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyPoint-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 21
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

lifeSafetyZone Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics associated with an arbitrary group of BACnet Life Safety Point and Life Safety Zone objects in fire, life safety and security applications. The condition of a Life Safety Zone object is represented by a mode and a state. Mode changes determine the object's inner logic and, consequently, influence the evaluation of the state. Typically, the operating mode would be under operator control. The state of the object represents the condition of the controller according to the logic internal to the device. The implementation of the logic applied to such controllers to determine the various possible states is a local matter. Typical

applications of the Life Safety Zone object include fire zones, panel zones, detector lines, extinguishing controllers, remote transmission controllers, etc. Similar objects can be identified in security control panels.

Objects of type lifeSafetyZone have the following properties:

- **acceptedModes**
- **ackedTransitions**
- **alarmValues**
- **description**
- **deviceType**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultValues**
- **lifeSafetyAlarmValues**
- **maintenanceRequired**
- **memberOf**
- **mode**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **operationExpected**
- **outOfService**
- **presentValue**
- **profileLocation**
- **profileName**
- **propertyList**

- **reliability**
- **reliabilityEvaluationInhibit**
- **silenced**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **trackingValue**
- **valueSource**
- **zoneMembers**

lifeSafetyZone.acceptedModes

According to the BACnet protocol documentation:

Specifies all values the Mode property accepts when written to.

Property *acceptedModes* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.acceptedModes

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 175

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLifeSafetyMode>**.

lifeSafetyZone.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *lifeSafetyZone* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 22
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyZone.alarmValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-OFFNORMAL event is generated.

Property *alarmValues* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.alarmValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 7

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLifeSafetyState>**.

lifeSafetyZone.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

lifeSafetyZone.deviceType

According to the BACnet protocol documentation:

String describing the physical device connected to the analog input. It will typically be used to describe the type of sensor attached to the analog input.

Property *deviceType* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.deviceType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 31

Tag Value

A string value.

lifeSafetyZone.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `lifeSafetyZone` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyZone.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property `eventAlgorithmInhibitRef` of object `lifeSafetyZone` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

lifeSafetyZone.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyZone.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *lifeSafetyZone* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 22
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyZone.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *lifeSafetyZone* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 22
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

lifeSafetyZone.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *lifeSafetyZone* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 22
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

lifeSafetyZone.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

lifeSafetyZone.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *lifeSafetyZone* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 22
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

lifeSafetyZone.faultValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-FAULT event is generated.

Property *faultValues* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.faultValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 39

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLifeSafetyState>**.

lifeSafetyZone.lifeSafetyAlarmValues

According to the BACnet protocol documentation:

Specifies any states the PresentValue must equal before a TO-OFFNORMAL event is generated and event state LIFE_SAFETY_ALARM is entered.

Property *lifeSafetyAlarmValues* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.lifeSafetyAlarmValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 166

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLifeSafetyState>**.

lifeSafetyZone.maintenanceRequired

According to the BACnet protocol documentation:

Indicates that maintenance is required for one or more of the life safety points that are members of this zone.

Property *maintenanceRequired* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.maintenanceRequired

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 158

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyZone.memberOf

According to the BACnet protocol documentation:

Indicates those LifeSafetyZone objects of which this LifeSafetyPoint object is considered to be a zone member.

Property *memberOf* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.memberOf

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 159

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectReference>**.

lifeSafetyZone.mode

According to the BACnet protocol documentation:

Conveys the desired operating mode for the Life Safety Point object.

Property *mode* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.mode

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 160

Tag Value

An integer representing one of the options of enumeration **<BACnetLifeSafetyMode>**.

lifeSafetyZone.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

lifeSafetyZone.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

lifeSafetyZone.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

lifeSafetyZone.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

lifeSafetyZone.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

lifeSafetyZone.operationExpected

According to the BACnet protocol documentation:

Specifies the next operation expected by this object to handle a specific life safety situation.

Property *operationExpected* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.operationExpected

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 161

Tag Value

An integer representing one of the options of enumeration **<BACnetLifeSafetyOperation>**.

lifeSafetyZone.outOfService

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyZone.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *lifeSafetyZone* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

An integer representing one of the options of enumeration **<BACnetLifeSafetyState>**.

lifeSafetyZone.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

lifeSafetyZone.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

lifeSafetyZone.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

lifeSafetyZone.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

lifeSafetyZone.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyZone.silenced

According to the BACnet protocol documentation:

Indicates whether the most recently occurring transition for this object that has produced an audible orr visual indication has been silenced by the receipt of a LifeSafetyOperation service request or a local process.

Property *silenced* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.silenced

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 163

Tag Value

An integer representing one of the options of enumeration **<BACnetSilencedState>**.

lifeSafetyZone.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *lifeSafetyZone* can be read and written using a block tag with 4 elements.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 22
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lifeSafetyZone.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

lifeSafetyZone.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

lifeSafetyZone.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

lifeSafetyZone.trackingValue

According to the BACnet protocol documentation:

Reflects the non-latched state of the Life Safety Point object.

Property *trackingValue* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.trackingValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 164

Tag Value

An integer representing one of the options of enumeration **<BACnetLifeSafetyState>**.

lifeSafetyZone.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

lifeSafetyZone.zoneMembers

According to the BACnet protocol documentation:

Indicates which LifeSafetyPoint and LifeSafetyZone objects are members of the zone represented by this object.

Property *zoneMembers* of object *lifeSafetyZone* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lifeSafetyZone-<objectAddress>.zoneMembers

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 22
N3:	<objectAddress>
N4:	<property> = 165

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectReference>**.

lift Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a lift. As there could be multiple car doors on a lift car, there could also be up to the same number of landing doors for the lift car at each floor. Normally, a landing door is driven by the car door as landing doors are not powered. So the status of a lift car door also reflects the status of the corresponding landing door at a particular floor.

Objects of type lift have the following properties:

- **ackedTransitions**
- **assignedLandingCalls**
- **carAssignedDirection**
- **carDoorCommand**
- **carDoorStatus**
- **carDoorText**
- **carDoorZone**
- **carDriveStatus**
- **carLoad**
- **carLoadUnits**
- **carMode**
- **carMovingDirection**
- **carPosition**
- **description**
- **elevatorGroup**

- **energyMeter**
- **energyMeterRef**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultSignals**
- **floorText**
- **groupId**
- **higherDeck**
- **installationId**
- **landingDoorStatus**
- **lowerDeck**
- **makingCarCall**
- **nextStoppingFloor**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **passengerAlarm**
- **profileLocation**
- **profileName**
- **propertyList**
- **registeredCarCall**
- **reliability**
- **reliabilityEvaluationInhibit**

- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**

lift.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *lift* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 59
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lift.assignedLandingCalls

According to the BACnet protocol documentation:

Represent the current landing calls and their direction for the lift represented by this object. Each array element represents the list of assigned landing calls for the door of the car assigned to this array element.

Property *assignedLandingCalls* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.assignedLandingCalls

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 447

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAssignedLandingCalls>**.

lift.carAssignedDirection

According to the BACnet protocol documentation:

Represents the direction the lift is assigned to move, based on current car calls.

Property *carAssignedDirection* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.carAssignedDirection

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 448

Tag Value

An integer representing one of the options of enumeration **<BACnetLiftCarDirection>**.

lift.carDoorCommand

According to the BACnet protocol documentation:

Indicates the last pending car door commands written to this property. Writing to this property is equivalent to a passenger requesting that the respective car door be opened or closed. Each array element represents the last pending car door command for the door of the car assigned to this array element. Once the respective car door command is executed or no longer applicable, e.g., the car is now moving, the respective array element shall revert to NONE.

Property *carDoorCommand* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.carDoorCommand

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 449

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLiftCarDoorCommand>**.

lift.carDoorStatus

According to the BACnet protocol documentation:

Indicates the status of the doors on the car. Each array element indicates the status of the car door assigned to this array element.

Property *carDoorStatus* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.carDoorStatus

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 450

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDoorStatus>**.

lift.carDoorText

According to the BACnet protocol documentation:

Represents the descriptions or names for the doors of the lift car. Each array element represents the description or name for the door of the car assigned to this array element.

Property *carDoorText* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.carDoorText

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 451

Tag Value

A character string containing an XML with a value of type **<SequenceOfCharacterString>**.

lift.carDoorZone

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the car is in the door zone, the region near the landing where the door is permitted to start opening.

Property *carDoorZone* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.carDoorZone

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 452

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lift.carDriveStatus

According to the BACnet protocol documentation:

Indicates the current status of the lift's motor drive system.

Property *carDriveStatus* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.carDriveStatus

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 453

Tag Value

An integer representing one of the options of enumeration **<BACnetLiftCarDriveStatus>**.

lift.carLoad

According to the BACnet protocol documentation:

Indicates the load in the car, both passengers and goods. The value of the Car_Load property shall be in units as indicated by the Car_Load_Units property.

Property *carLoad* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.carLoad

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 454

Tag Value

A 32-bit floating point number.

lift.carLoadUnits

According to the BACnet protocol documentation:

Indicates the measurement units of the Car_Load property.

Property *carLoadUnits* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.carLoadUnits

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 455

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

lift.carMode

According to the BACnet protocol documentation:

Indicates the current operational mode of the car.

Property *carMode* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.carMode

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 456

Tag Value

An integer representing one of the options of enumeration **<BACnetLiftCarMode>**.

lift.carMovingDirection

According to the BACnet protocol documentation:

Represents whether or not this lift's car is moving, and if so, in which direction.

Property *carMovingDirection* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.carMovingDirection

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 457

Tag Value

An integer representing one of the options of enumeration **<BACnetLiftCarDirection>**.

lift.carPosition

According to the BACnet protocol documentation:

Indicates the universal floor number of this lift's car position.

Property *carPosition* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.carPosition

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 458

Tag Value

An 8-bit integer value ranging from 0 to 255.

lift.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

lift.elevatorGroup

According to the BACnet protocol documentation:

References the Elevator Group object whose Group_Members property contains a reference to this Lift object. If there is no such Elevator Group object, this property shall contain an object instance of 4194303.

Property *elevatorGroup* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.elevatorGroup

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 459

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

lift.energyMeter

According to the BACnet protocol documentation:

Indicates the accumulated energy consumption by the lift. The units shall be kilowatt-hours. When this value reaches 99999 kWh, it shall wrap to a value near zero; the particular value to which it wraps is a local matter. If the Energy_Meter_Ref property is present and initialized (contains an instance other than 4194303), then the Energy_Meter property, if present, shall contain a value of 0.0.

Property *energyMeter* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.energyMeter

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 460

Tag Value

A 32-bit floating point number.

lift.energyMeterRef

According to the BACnet protocol documentation:

References the object which indicates the accumulated energy consumption by the lift.

Property *energyMeterRef* of object *lift* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.energyMeterRef
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 59
B3:	<objectAddress>
B4:	<property> = 461
Size:	2 elements

Block Elements

Index	Name	Type
0	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>--<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.
1	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>--<Address>' (ex: 'analogOutput-345').

lift.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `lift` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lift.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property `Event_Algorithm_Inhibit`. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type `BACnetBinaryPV` or `BOOLEAN`.

Property `eventAlgorithmInhibitRef` of object `lift` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '<index>)' can be used on array properties.

lift.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, `Event_State` shall be `NORMAL`, and the properties `Acked_Transitions`, `Event_Time_Stamps`, and `Event_Message_Texts` shall be equal to their respective initial conditions.

Property `eventDetectionEnable` of object `lift` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lift.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *lift* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 59
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lift.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *lift* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 59
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

lift.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *lift* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 59
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

lift.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

lift.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *lift* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 59
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

lift.faultSignals

According to the BACnet protocol documentation:

Represents a list of values that indicates fault conditions of the lift.

Property *faultSignals* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.faultSignals

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 463

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLiftFault>**.

lift.floorText

According to the BACnet protocol documentation:

Represents the descriptions or names for the floors. The universal floor number serves as an index into this array. The size of this array shall match the highest universal floor number served by this lift.

Property *floorText* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.floorText

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 464

Tag Value

A character string containing an XML with a value of type **<SequenceOfCharacterString>**.

lift.groupId

According to the BACnet protocol documentation:

Represents the identification number for the group of lifts or escalators represented by this object. This identification number shall be unique for the groups in this machine room, but might not be otherwise unique in the building.

Property *groupId* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.groupId

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 465

Tag Value

An 8-bit integer value ranging from 0 to 255.

lift.higherDeck

According to the BACnet protocol documentation:

References the Lift object that is representing the car deck above the car deck represented by this object. If this property is present, and there is no higher deck, then the object instance shall be 4194303.

Property *higherDeck* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.higherDeck

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 468

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

lift.installationId

According to the BACnet protocol documentation:

Represents the identification number for the lift represented by this object. This identification number shall be unique for the lift in this group, but might not be otherwise unique for other lifts in the machine room or the building.

Property *installationId* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.installationId

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 469

Tag Value

An 8-bit integer value ranging from 0 to 255.

lift.landingDoorStatus

According to the BACnet protocol documentation:

Represents the status of the landing doors on the floors served by this lift. Each element of this array represents the list of landing doors for the door of the car assigned to this array element. A landing door status includes the universal floor number and the currently active door status for the landing door.

Property *landingDoorStatus* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.landingDoorStatus

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 472

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLandingDoorStatus>**.

lift.lowerDeck

According to the BACnet protocol documentation:

References the Lift object that is representing the car deck below the car deck represented by this object. If this property is present, and there is no lower deck, then the object instance shall be 4194303.

Property *lowerDeck* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.lowerDeck

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 473

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

lift.makingCarCall

According to the BACnet protocol documentation:

Indicates the last car calls written to this property. Writing to this property is equivalent to a passenger requesting that the car stop at the designated floor. Each array element represents the last car call written to this property for the door of the car assigned to this array element. If no car call has been written to an array element, the array element shall indicate a value of zero.

Property *makingCarCall* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.makingCarCall

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 475

Tag Value

A character string containing an XML with a value of type **<SequenceOfUnsigned8>**.

lift.nextStoppingFloor

According to the BACnet protocol documentation:

Indicates the universal floor number where the car will stop next when underway. If the car is not in motion, this property indicates the current universal floor number.

Property *nextStoppingFloor* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.nextStoppingFloor

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 476

Tag Value

An 8-bit integer value ranging from 0 to 255.

lift.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

lift.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

lift.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

lift.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

lift.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

lift.outOfService

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lift.passengerAlarm

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the passenger alarm has been activated.

Property *passengerAlarm* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.passengerAlarm

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 478

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lift.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

lift.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

lift.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

lift.registeredCarCall

According to the BACnet protocol documentation:

Represents the lists of currently registered car calls (requests to stop at particular floors using a particular door) for this lift. Each array element represents the list of universal floor numbers for which calls are registered for the door of the car assigned to this array element.

Property *registeredCarCall* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.registeredCarCall

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 480

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLiftCarCallList>**.

lift.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

lift.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lift.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *lift* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 59
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lift.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

lift.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *lift* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

lift.timeDelayNormal

According to the BACnet protocol documentation:

Contains the `pTimeDelayNormal` parameter for the object's event algorithm.

Property `timeDelayNormal` of object `lift` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lift-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 59
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

lightingOutput Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a lighting output and includes dedicated functionality specific to lighting control that would otherwise require explicit programming. The lighting output is analog in nature. The physical output level, or non-normalized range, is specified as the linearized percentage (0..100%) of the possible light output range with 0.0% being off, 1.0% being dimmest, and 100.0% being brightest. The actual range represents the subset of physical output levels defined by `Min_Actual_Value` and `Max_Actual_Value` (or 1.0 to 100.0% if these properties are not present). The normalized range is always 0.0 to 100.0% where 1.0% = bottom of the actual range and 100.0% = top of the actual range. All 0.0% to 100.0% properties of the Lighting Output object shall use the normalized range except for `Min_Actual_Value` and `Max_Actual_Value`. If `Min_Actual_Value` and `Max_Actual_Value` are not present, then the normalized and non-normalized ranges shall be the same.

Objects of type `lightingOutput` have the following properties:

- **blinkWarnEnable**
- **commandTimeArray**
- **covIncrement**
- **currentCommandPriority**
- **defaultFadeTime**
- **defaultRampRate**

- **defaultStepIncrement**
- **description**
- **egressActive**
- **egressTime**
- **feedbackValue**
- **inProgress**
- **instantaneousPower**
- **lastCommandTime**
- **lightingCommand**
- **lightingCommandDefaultPriority**
- **maxActualValue**
- **minActualValue**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **power**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **statusFlags**
- **tags**
- **trackingValue**
- **transition**
- **valueSource**
- **valueSourceArray**

lightingOutput.blinkWarnEnable

According to the BACnet protocol documentation:

Specifies whether a blink-warn is executed (TRUE) or not (FALSE) when a WARN, WARN_RELINQUISH, or WARN_OFF command is written to the Lighting_Command property or one of the special values is written to the Present_Value. When this property is FALSE and a warn operation is written, a blink-warn notification shall not occur, and the effect of the operation shall occur immediately without an egress delay.

Property *blinkWarnEnable* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.blinkWarnEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 373

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lightingOutput.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *lightingOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 54
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

lightingOutput.covIncrement

According to the BACnet protocol documentation:

Specifies the minimum change in presentValue that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

Property *covIncrement* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.covIncrement

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 22

Tag Value

A 32-bit floating point number.

lightingOutput.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

lightingOutput.defaultFadeTime

According to the BACnet protocol documentation:

Indicates the amount of time in milliseconds over which changes to the normalized value reflected in the Tracking_Value property of the lighting output shall occur when the Lighting_Command property is written with a fade request that does not include a fade-time value. The range of allowable fade-time values is 100 ms to 86400000 ms (1 day) inclusive.

Property *defaultFadeTime* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.defaultFadeTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 374

Tag Value

A 32-bit unsigned integer.

lightingOutput.defaultRampRate

According to the BACnet protocol documentation:

Indicates the rate in percent-per-second at which changes to the normalized value reflected in the Tracking_Value property of the lighting output shall occur when the Lighting_Command property is written with a ramp request that does not include a ramp-rate value. The range of allowable ramp-rate values is 0.1 %/s to 100.0 %/s inclusive.

Property *defaultRampRate* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.defaultRampRate

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 375

Tag Value

A 32-bit floating point number.

lightingOutput.defaultStepIncrement

According to the BACnet protocol documentation:

Indicates the amount to be added to the Tracking_Value when the Lighting_Command property is written with a step request that does not include a step-increment value. The range of allowable values is 0.1% to 100.0% inclusive.

Property *defaultStepIncrement* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.defaultStepIncrement

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 376

Tag Value

A 32-bit floating point number.

lightingOutput.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

lightingOutput.egressActive

According to the BACnet protocol documentation:

Shall be TRUE whenever the Egress_Time for a WARN_RELINQUISH or WARN_OFF lighting operation is in effect and FALSE otherwise.

Property *egressActive* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.egressActive

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 386

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lightingOutput.egressTime

According to the BACnet protocol documentation:

specifies the egress time in seconds when a WARN_RELINQUISH or WARN_OFF is written to the Lighting_Command property or when the special values -2.0 or -3.0 are written to the Present_Value property. The egress time is the time for which the light level is held at its current level before it is relinquished or set to 0.0%.

Property *egressTime* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.egressTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 377

Tag Value

A 32-bit unsigned integer.

lightingOutput.feedbackValue

According to the BACnet protocol documentation:

Indicates the status of a feedback value from which the presentValue must differ before an event is generated.

Property *feedbackValue* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.feedbackValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 40

Tag Value

A 32-bit floating point number.

lightingOutput.inProgress

According to the BACnet protocol documentation:

Indicates processes in the lighting output object that may cause the Tracking_Value and Present_Value to differ temporarily.

Property *inProgress* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.inProgress

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 378

Tag Value

An integer representing one of the options of enumeration **<BACnetLightingInProgress>**.

lightingOutput.instantaneousPower

According to the BACnet protocol documentation:

Conveys the nominal power consumption of the load(s) controlled by this object at this moment. The units shall be kilowatts.

Property *instantaneousPower* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.instantaneousPower

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 379

Tag Value

A 32-bit floating point number.

lightingOutput.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

lightingOutput.lightingCommand

According to the BACnet protocol documentation:

Used to request special lighting commands with specific behaviors. Lighting_Command is written with compound values that specify particular lighting operations. When a lighting operation is written to the Lighting_Command property, the effect of that operation is written to the Present_Value at the priority level specified by the priority field. If the priority field is not included with the command, the priority specified in Lighting_Command_Default_Priority shall be used. Some lighting operations require additional parameters. These are provided by optional fields of the BACnetLightingCommand value.

Property *lightingCommand* of object *lightingOutput* can be read and written using a block tag with 6 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.lightingCommand
Size	6 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 54
B3:	<objectAddress>
B4:	<property> = 380
Size:	6 elements

Block Elements

Index	Name	Type
0	Operation	An integer representing one of the options of enumeration <BACnetLightingOperation> .
1	TargetLevel	A 32-bit floating point number. OBS: this is an optional element, a <null> value indicates the element is not present.
2	RampRate	A 32-bit floating point number. OBS: this is an optional element, a <null> value indicates the element is not present.
3	StepIncrement	A 32-bit floating point number. OBS: this is an optional element, a <null> value indicates the element is not present.
4	FadeTime	A 32-bit unsigned integer. OBS: this is an optional element, a <null> value indicates the element is not present.
5	Priority	A 32-bit unsigned integer. OBS: this is an optional element, a <null> value indicates the element is not present.

lightingOutput.lightingCommandDefaultPriority

According to the BACnet protocol documentation:

Specifies a write priority of 1 to 16 that indicates the element of the Priority_Array controlled by the Lighting_Command property when the BACnetLightingCommand priority field is absent. The priority value 6 shall not be used for this property.

Property *lightingCommandDefaultPriority* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.lightingCommandDefaultPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 381

Tag Value

A 32-bit unsigned integer.

lightingOutput.maxActualValue

According to the BACnet protocol documentation:

Specifies the physical output level that corresponds to a Present_Value of 100.0%. Changing Max_Actual_Value to a value less than Min_Actual_Value shall force Min_Actual_Value to become equal to Max_Actual_Value. Max_Actual_Value shall always be a positive number in the range 1.0% to 100.0%.

Property *maxActualValue* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.maxActualValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 382

Tag Value

A 32-bit floating point number.

lightingOutput.minActualValue

According to the BACnet protocol documentation:

Specifies the physical output level that corresponds to a Present_Value of 1.0%. Changing Min_Actual_Value to a value greater than Max_Actual_Value shall force Max_Actual_Value to become equal to Min_Actual_Value. Min_Actual_Value shall always be a positive number in the range 1.0% to 100.0%.

Property *minActualValue* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.minActualValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 383

Tag Value

A 32-bit floating point number.

lightingOutput.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

lightingOutput.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

lightingOutput.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

lightingOutput.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lightingOutput.power

According to the BACnet protocol documentation:

Indicates the nominal power consumption of the load(s) controlled by this object when the light level is 100.0% of the non-normalized range. The units shall be kilowatts.

Property *power* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.power

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 384

Tag Value

A 32-bit floating point number.

lightingOutput.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit floating point number.

lightingOutput.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *lightingOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 54
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
1	elm01	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
2	elm02	One of the following data types:

lightingOutput.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

lightingOutput.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

lightingOutput.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

lightingOutput.reliability

According to the BACnet protocol documentation:

Indicates whether the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

lightingOutput.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lightingOutput.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A 32-bit floating point number.

lightingOutput.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *lightingOutput* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 54
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

lightingOutput.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

lightingOutput.trackingValue

According to the BACnet protocol documentation:

Reflects the non-latched state of the Life Safety Point object.

Property *trackingValue* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.trackingValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 164

Tag Value

A 32-bit floating point number.

lightingOutput.transition

According to the BACnet protocol documentation:

Specifies how a change in the Present_Value transitions from the current level to the target level. A transition comes into effect when the Present_Value is directly commanded or when the current highest priority value has been relinquished. Writing the Lighting commands FADE_TO, RAMP_TO, STEP_ON, STEP_OFF, STEP_UP, or STEP_DOWN shall ignore the Transition property. The transition may be one of NONE, FADE, or RAMP. The transition NONE causes the Present_Value to immediately be set to the target level when the highest priority value has been relinquished. If this property does not exist, then the transition type shall be assumed to be NONE. FADE or RAMP transitions allow a smooth transition of the lighting level when the Present_Value changes. A FADE transition executes a fade operation from the Tracking_Value to the target level using the fade time specified in Default_Fade_Time. A RAMP transition executes a ramp operation from the Tracking_Value to the target level using the ramp rate specified in Default_Ramp_Rate. When a transition results in an operation that may cause the Tracking_Value to differ from the Present_Value, then the In_Progress property shall be set to the value that reflects the operation in progress.

Property *transition* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.transition

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 385

Tag Value

An integer representing one of the options of enumeration **<BACnetLightingTransition>**.

lightingOutput.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *lightingOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 54
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

lightingOutput.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *lightingOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	lightingOutput-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 54
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types:

		• A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value.

- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

loadControl Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a mechanism for controlling load requirements. A BACnet device can use a Load Control object to allow external control over the shedding of a load that it controls. The mechanisms by which the loads are shed are not visible to the BACnet client. One or more objects may be used in the device to allow independent control over different sub-loads. The Load Control object may also be used in a hierarchical fashion to control other Load Control objects in other BACnet devices.

Objects of type loadControl have the following properties:

- **ackedTransitions**
- **actualShedLevel**
- **description**
- **dutyWindow**
- **enable**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **expectedShedLevel**
- **fullDutyBaseline**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **presentValue**
- **profileLocation**

- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **requestedShedLevel**
- **shedDuration**
- **shedLevelDescriptions**
- **shedLevels**
- **startTime**
- **stateDescription**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **valueSource**

loadControl.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *loadControl* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 28
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loadControl.actualShedLevel

According to the BACnet protocol documentation:

Indicates the actual amount of power being shed in response to a load shed request. When the object is in the SHED_INACTIVE state, this value shall be equal to the default value of Requested_Shed_Level. After Start_Time plus Duty_Window has elapsed, this value shall be the actual shed amount as calculated based on the average value over the previous duty window. The units for Actual_Shed_Level are the same as the units for Requested_Shed_Level.

Property *actualShedLevel* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.actualShedLevel

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 212

Tag Value

One of the following data types:

- A 32-bit unsigned integer.
- A character string containing an XML with a value of type **<Unsigned>**.
- A 32-bit floating point number.

loadControl.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

loadControl.dutyWindow

According to the BACnet protocol documentation:

Indicates the time window used for load shed accounting. The units for Duty_Window are minutes. Duty_Window is used for performance measurement or compliance purposes. The average power consumption across a duty window must be less than or equal to the requested reduced consumption. It is a local matter whether this window is fixed or sliding. The first Duty_Window begins at Start_Time. If a shed request is received with no value written to this property, Duty_Window shall be set to some pre-agreed upon value. If a load control command has been issued, and execution of the command has completed, Duty_Window shall be reset by the device to this pre-agreed value.

Property *dutyWindow* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.dutyWindow

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 213

Tag Value

A 32-bit unsigned integer.

loadControl.enable

According to the BACnet protocol documentation:

Indicates and controls whether (TRUE) or not (FALSE) logging of events is enabled. Logging occurs if and only if Enable is TRUE, Local_Time is on or after Start_Time, and Local_Time is before Stop_Time. If Start_Time contains an unspecified datetime, then it shall be considered equal to 'the start of time'. If Stop_Time contains an unspecified datetime, then it shall be considered equal to 'the end of time'. Log records of type log-status are recorded without regard to the value of the Enable property.

Property *enable* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.enable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 133

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loadControl.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loadControl.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

loadControl.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loadControl.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *loadControl* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 28
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loadControl.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *loadControl* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 28
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

loadControl.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *loadControl* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 28
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

loadControl.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

loadControl.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *loadControl* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 28
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

loadControl.expectedShedLevel

According to the BACnet protocol documentation:

Indicates the amount of power that the object expects to be able to shed in response to a load shed request. When the object is in the SHED_INACTIVE state, this value shall be equal to the default value of Requested_Shed_Level. When a shed request is pending or active, Expected_Shed_Level shall be equal to the shed level the object expects to be able to achieve at Start_Time. Expected_Shed_Level allows a client (e.g., a master-level Load Control object) to determine if a pending shed request needs to be modified in order to achieve the requested shed level, in the event that Expected_Shed_Level is less than the Requested_Shed_Level. The units for Expected_Shed_Level are the same as the units for Requested_Shed_Level.

Property *expectedShedLevel* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.expectedShedLevel

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 214

Tag Value

One of the following data types:

- A 32-bit unsigned integer.
- A character string containing an XML with a value of type **<Unsigned>**.
- A 32-bit floating point number.

loadControl.fullDutyBaseline

According to the BACnet protocol documentation:

Indicates the baseline power consumption value for the sheddable load controlled by this object, if a fixed baseline is used. Shed requests may be made with respect to this baseline, that is, to "percent of baseline" and "amount off baseline". The units of Full_Duty_Baseline are kilowatts.

Property *fullDutyBaseline* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.fullDutyBaseline

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 215

Tag Value

A 32-bit floating point number.

loadControl.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

loadControl.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

loadControl.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

loadControl.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

loadControl.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

loadControl.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

An integer representing one of the options of enumeration **<BACnetShedState>**.

loadControl.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

loadControl.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

loadControl.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

loadControl.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

loadControl.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loadControl.requestedShedLevel

According to the BACnet protocol documentation:

Indicates the desired load shedding. If the choice for *Requested_Shed_Level* is *PERCENT*, the value of *Requested_Shed_Level* is interpreted as a requested percentage of Full Duty to which the device is to attempt to reduce its load. The determination of the Full Duty rating (or some alternative baseline power usage) is a local matter. It may be determined from the *Full_Duty_Baseline* property, if present. If the choice for *Requested_Shed_Level* is *LEVEL*, the value of *Requested_Shed_Level* is used to set a preconfigured level of load shedding. The Load Control object's available shed actions are described by the *Shed_Level_Descriptions* array and are mapped to the BACnet visible values of *Requested_Shed_Level* by the *Shed_Levels* array. The *SHED_INACTIVE* state shall always be represented by the value 0, which is not represented in the *Shed_Levels* or *Shed_Level_Descriptions* arrays. If *Requested_Shed_Level* choice is *AMOUNT*, the value of *Requested_Shed_Level* shall be interpreted as an amount, in kilowatts, by which to reduce power usage. Load Control objects are required to support the *LEVEL* choice. Support for the *PERCENT* and *AMOUNT* choices is optional. This allows a master to be guaranteed the ability to write to the Load Control object by using the *LEVEL* choice. If a load control command has been issued, and execution of the command has completed, *Requested_Shed_Level* shall be reset to the default value appropriate to the choice of *Requested_Shed_Level* used for the last command.

Property *requestedShedLevel* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.requestedShedLevel

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 218

Tag Value

One of the following data types:

- A 32-bit unsigned integer.
- A character string containing an XML with a value of type **<Unsigned>**.
- A 32-bit floating point number.

loadControl.shedDuration

According to the BACnet protocol documentation:

Indicates the duration of the load shed action, starting at Start_Time. The units for Shed_Duration are minutes. If no shed request is pending or active, Shed_Duration shall be zero. If a load control command has been issued, and execution of the command has completed, Shed_Duration shall be reset by the device to zero.

Property *shedDuration* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.shedDuration

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 219

Tag Value

A 32-bit unsigned integer.

loadControl.shedLevelDescriptions

According to the BACnet protocol documentation:

Represents a description of the shed levels that the Load Control object can take on. This allows a local configuration tool to provide to a user an understanding of what each shed level in this Load Control object's load shedding algorithm will do. The level at which each shed action will occur can then be configured by writing to the Shed_Levels property.

Property *shedLevelDescriptions* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.shedLevelDescriptions

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 220

Tag Value

A character string containing an XML with a value of type **<SequenceOfCharacterString>**.

loadControl.shedLevels

According to the BACnet protocol documentation:

An array of unsigned integers representing the shed levels for the LEVEL choice of BACnetShedLevel that have meaning for this particular Load Control object. The array shall be ordered by increasing shed amount. When commanded with the LEVEL choice, the Load Control object shall take a shedding action described by the corresponding element in the

Shed_Level_Descriptions array. If the Load Control object is commanded to go to a level that is not in the Shed_Levels array, it shall go to the Shed_Level whose entry in the Shed_Levels array has the nearest numerically lower value. The elements of the array are required to be writable, allowing local configuration of how this Load Control object will participate in load shedding for the facility. This array is not required to be resizable through BACnet write services. The size of this array shall be equal to the size of the Shed_Level_Descriptions array. The behavior of this object when the Shed_Levels array contains duplicate entries is a local matter.

Property *shedLevels* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.shedLevels

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 221

Tag Value

A character string containing an XML with a value of type **<SequenceOfUnsigned>**.

loadControl.startTime

According to the BACnet protocol documentation:

Specifies the date and time at or after which logging shall be enabled by this property. If this property contains an unspecified datetime, then the conditions for logging to be enabled by Start_Time shall be ignored. If Start_Time specifies a date and time after Stop_Time, then logging shall be disabled. This property shall be writable if present. When Start_Time is reached, the value of the Enable property is not changed.

Property *startTime* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.startTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 142

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

loadControl.stateDescription

According to the BACnet protocol documentation:

Contains a string of printable characters whose content is not restricted. The State_Description provides additional information for human operators about the shed state of the Load Control object.

Property *stateDescription* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.stateDescription

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 222

Tag Value

A string value.

loadControl.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *loadControl* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 28
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loadControl.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

loadControl.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

loadControl.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

loadControl.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *loadControl* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loadControl-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 28
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

loop Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of any form of feedback control loop. Flexibility is achieved by providing three independent gain constants with no assumed values for units. The appropriate gain units are determined by the details of the control algorithm, which is a local matter.

Objects of type loop have the following properties:

- **ackedTransitions**
- **action**
- **bias**
- **controlledVariableReference**
- **controlledVariableUnits**
- **controlledVariableValue**
- **covIncrement**
- **deadband**
- **derivativeConstant**
- **derivativeConstantUnits**
- **description**
- **errorLimit**

- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **integralConstant**
- **integralConstantUnits**
- **lowDiffLimit**
- **manipulatedVariableReference**
- **maximumOutput**
- **minimumOutput**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **outputUnits**
- **presentValue**
- **priorityForWriting**
- **profileLocation**
- **profileName**
- **propertyList**
- **proportionalConstant**
- **proportionalConstantUnits**
- **reliability**
- **reliabilityEvaluationInhibit**
- **setpoint**
- **setpointReference**

- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **updateInterval**

loop.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *loop* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 12
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loop.action

According to the BACnet protocol documentation:

Specifies an array of 'action lists'.

Property *action* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.action

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 2

Tag Value

An integer representing one of the options of enumeration **<BACnetAction>**.

loop.bias

According to the BACnet protocol documentation:

The bias value used by the loop algorithm expressed in units of the outputUnits property.

Property *bias* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.bias

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 14

Tag Value

A 32-bit floating point number.

loop.controlledVariableReference

According to the BACnet protocol documentation:

Identifies the property used to set the controlledVariableValue property of the Loop object.

Property *controlledVariableReference* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.controlledVariableReference

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 19

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

loop.controlledVariableUnits

According to the BACnet protocol documentation:

Indicates the engineering units for the controlledVariableValue property of this object.

Property *controlledVariableUnits* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.controlledVariableUnits

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 20

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

loop.controlledVariableValue

According to the BACnet protocol documentation:

The value of the property of the object referenced by the controlledVariableReference property.

Property *controlledVariableValue* of object *loop* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.controlledVariableValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 21

Tag Value

A 32-bit floating point number.

loop.covIncrement

According to the BACnet protocol documentation:

Specifies the minimum change in presentValue that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

Property *covIncrement* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.covIncrement

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 22

Tag Value

A 32-bit floating point number.

loop.deadband

According to the BACnet protocol documentation:

Specifies a range between the `highLimit` and `lowLimit` properties, which the `presentValue` must remain within for a `TO-NORMAL` event to be generated.

Property `deadband` of object `loop` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.deadband

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 25

Tag Value

A 32-bit floating point number.

loop.derivativeConstant

According to the BACnet protocol documentation:

The value of the derivative gain parameter used by the loop algorithm.

Property `derivativeConstant` of object `loop` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.derivativeConstant

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 26

Tag Value

A 32-bit floating point number.

loop.derivativeConstantUnits

According to the BACnet protocol documentation:

Indicates the engineering units of the derivativeConstant property of this object.

Property *derivativeConstantUnits* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.derivativeConstantUnits

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 27

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

loop.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

loop.errorLimit

According to the BACnet protocol documentation:

Conveys the absolute magnitude that the difference between the Setpoint and controlledVariableValue (the Error) must exceed before a TO-OFFNORMAL event is generated.

Property *errorLimit* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.errorLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 34

Tag Value

A 32-bit floating point number.

loop.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loop.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

loop.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loop.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *loop* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 12
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loop.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property `eventMessageTexts` of object `loop` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 12
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

loop.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property `eventMessageTextsConfig` of object `loop` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 12
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

loop.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

loop.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *loop* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 12
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. - A null value means the timestamp is empty/not set; - A date/time value representing a precise date and time; - An integer defining the timestamp sequence number; - A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. - A null value means the timestamp is empty/not set; - A date/time value representing a precise date and time; - An integer defining the timestamp sequence number; - A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. - A null value means the timestamp is empty/not set; - A date/time value representing a precise date and time; - An integer defining the timestamp sequence number; - A character string containing an XML with a value of type <BACnetTimeStamp>.

loop.integralConstant

According to the BACnet protocol documentation:

The value of the integral gain parameter used by the loop algorithm.

Property *integralConstant* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.integralConstant

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 49

Tag Value

A 32-bit floating point number.

loop.integralConstantUnits

According to the BACnet protocol documentation:

Indicates the engineering units of the integralConstant property of this object.

Property *integralConstantUnits* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.integralConstantUnits

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 50

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

loop.lowDiffLimit

According to the BACnet protocol documentation:

Indicates the pLowDiffLimit parameter for the object's FLOATING_LIMIT event algorithm if it has a value other than NULL.

Property *lowDiffLimit* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.lowDiffLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 390

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.

loop.manipulatedVariableReference

According to the BACnet protocol documentation:

Identifies a property of an object which receives the output of the control loop.

Property *manipulatedVariableReference* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.manipulatedVariableReference

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 60

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

loop.maximumOutput

According to the BACnet protocol documentation:

The maximum value of the presentValue property as limited by the PID loop algorithm.

Property *maximumOutput* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.maximumOutput

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 61

Tag Value

A 32-bit floating point number.

loop.minimumOutput

According to the BACnet protocol documentation:

The minimum value of the presentValue property as limited by the loop algorithm.

Property *minimumOutput* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.minimumOutput

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 68

Tag Value

A 32-bit floating point number.

loop.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

loop.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

loop.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

loop.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

loop.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

loop.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loop.outputUnits

According to the BACnet protocol documentation:

Indicates the engineering units for the output of this control loop.

Property *outputUnits* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.outputUnits

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 82

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

loop.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *loop* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit floating point number.

loop.priorityForWriting

According to the BACnet protocol documentation:

Provides a priority to be used by the command prioritization mechanism.

Property *priorityForWriting* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.priorityForWriting

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 88

Tag Value

A 32-bit unsigned integer.

loop.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

loop.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

loop.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

loop.proportionalConstant

According to the BACnet protocol documentation:

The value of the proportional gain parameter used by the loop algorithm.

Property *proportionalConstant* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.proportionalConstant

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 93

Tag Value

A 32-bit floating point number.

loop.proportionalConstantUnits

According to the BACnet protocol documentation:

The engineering units of the proportionalConstant property of this object.

Property *proportionalConstantUnits* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.proportionalConstantUnits

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 94

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

loop.reliability

According to the BACnet protocol documentation:

Indicates whether the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

loop.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loop.setpoint

According to the BACnet protocol documentation:

The value of the loop setpoint or of the property of the object referenced by the setpointReferenced property.

Property *setpoint* of object *loop* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.setpoint

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 108

Tag Value

A 32-bit floating point number.

loop.setpointReference

According to the BACnet protocol documentation:

List of references that has a length of zero or one, indicating if the setpoint for this control loop is fixed and contained in the Setpoint property, or a property of another object.

Property *setpointReference* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.setpointReference

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 109

Tag Value

A block where each element maps to a structure member.

- A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties. OBS: this is an optional element, a <null> value indicates the element is not present.

loop.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *loop* can be read and written using a block tag with 4 elements.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 12
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

loop.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

loop.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

loop.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

loop.updateInterval

According to the BACnet protocol documentation:

Indicates the maximum period of time between updates to the presentValue in hundredths of a second when the input is not overridden and not out-of-service.

Property *updateInterval* of object *loop* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	loop-<objectAddress>.updateInterval

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 12
N3:	<objectAddress>
N4:	<property> = 118

Tag Value

A 32-bit unsigned integer.

multiStateInput Object

According to the BACnet protocol documentation:

Defines a standardized object whose Present_Value represents the result of an algorithmic process within the BACnet device in which the object resides. The algorithmic process itself is a local matter and is not defined by the protocol. For example, the Present_Value or state of the Multi-state Input object may be the result of a logical combination of multiple binary inputs or the threshold of one or more analog inputs or the result of a mathematical computation. The Present_Value property is an integer number representing the state. The State_Text property associates a description with each state.

Objects of type multiStateInput have the following properties:

- **ackedTransitions**
- **alarmValues**
- **description**
- **deviceType**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultValues**
- **interfaceValue**
- **notificationClass**
- **notifyType**
- **numberOfStates**
- **objectIdentifier**

- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **stateText**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**

multiStateInput.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *multiStateInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 13
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateInput.alarmValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-OFFNORMAL event is generated.

Property *alarmValues* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.alarmValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 7

Tag Value

A character string containing an XML with a value of type **<SequenceOfUnsigned>**.

multiStateInput.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

multiStateInput.deviceType

According to the BACnet protocol documentation:

String describing the physical device connected to the analog input. It will typically be used to describe the type of sensor attached to the analog input.

Property *deviceType* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.deviceType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 31

Tag Value

A string value.

multiStateInput.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateInput.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

multiStateInput.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system

configuration and is not expected to change dynamically. When this property is *FALSE*, *Event_State* shall be *NORMAL*, and the properties *Acked_Transitions*, *Event_Time_Stamps*, and *Event_Message_Texts* shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to *FALSE* and any other value corresponds to *TRUE*.

multiStateInput.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *multiStateInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 13
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateInput.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *multiStateInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 13
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

multiStateInput.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *multiStateInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 13
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

multiStateInput.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

multiStateInput.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *multiStateInput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 13
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

multiStateInput.faultValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-FAULT event is generated.

Property *faultValues* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.faultValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 39

Tag Value

A character string containing an XML with a value of type **<SequenceOfUnsigned>**.

multiStateInput.interfaceValue

According to the BACnet protocol documentation:

Indicates the value, in engineering units, of the physical input. If the BACnet device is not capable of knowing the value of the physical input, then the value of this property shall be NULL.

Property *interfaceValue* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.interfaceValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 387

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

multiStateInput.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

multiStateInput.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

multiStateInput.numberOfStates

According to the BACnet protocol documentation:

Defines the number of states that the presentValue may have.

Property *numberOfStates* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.numberOfStates

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 74

Tag Value

A 32-bit unsigned integer.

multiStateInput.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

multiStateInput.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

multiStateInput.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

multiStateInput.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateInput.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *multiStateInput* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit unsigned integer.

multiStateInput.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

multiStateInput.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *fileName* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.fileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

multiStateInput.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

multiStateInput.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

multiStateInput.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateInput.stateText

According to the BACnet protocol documentation:

An array of character strings representing descriptions of all possible states of the presentValue.

Property *stateText* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.stateText

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 110

Tag Value

A character string containing an XML with a value of type **<SequenceOfCharacterString>**.

multiStateInput.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *multiStateInput* can be read and written using a block tag with 4 elements.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 13
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateInput.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

multiStateInput.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

multiStateInput.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *multiStateInput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateInput-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 13
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

multiStateOutput Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the desired state of one or more physical outputs or processes within the BACnet device in which the object resides. The actual functions associated with a specific state are a local matter and not specified by the protocol. For example, a particular state may represent the active/inactive condition of several physical outputs or perhaps the value of an analog output. The Present_Value property is an unsigned integer number representing the state. The State_Text property associates a description with each state.

Objects of type multiStateOutput have the following properties:

- **ackedTransitions**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **deviceType**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **feedbackValue**
- **interfaceValue**
- **lastCommandTime**
- **notificationClass**
- **notifyType**
- **numberOfStates**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**

- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **stateText**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **valueSource**
- **valueSourceArray**

multiStateOutput.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *multiStateOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 14
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateOutput.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *multiStateOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 14
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

		• A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
7	elm07	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
8	elm08	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

		• A character string containing an XML with a value of type <BACnetTimeStamp>.
9	elm09	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
10	elm10	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
11	elm11	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
12	elm12	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
13	elm13	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
14	elm14	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
15	elm15	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set;

- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

multiStateOutput.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

multiStateOutput.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

multiStateOutput.deviceType

According to the BACnet protocol documentation:

String describing the physical device connected to the analog input. It will typically be used to describe the type of sensor attached to the analog input.

Property *deviceType* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.deviceType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 31

Tag Value

A string value.

multiStateOutput.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateOutput.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

multiStateOutput.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system

configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamped, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateOutput.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *multiStateOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 14
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateOutput.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *multiStateOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 14
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

multiStateOutput.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *multiStateOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 14
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

multiStateOutput.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

multiStateOutput.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *multiStateOutput* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 14
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

multiStateOutput.feedbackValue

According to the BACnet protocol documentation:

Indicates the status of a feedback value from which the presentValue must differ before an event is generated.

Property *feedbackValue* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.feedbackValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 40

Tag Value

A 32-bit unsigned integer.

multiStateOutput.interfaceValue

According to the BACnet protocol documentation:

Indicates the value, in engineering units, of the physical input. If the BACnet device is not capable of knowing the value of the physical input, then the value of this property shall be NULL.

Property *interfaceValue* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.interfaceValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 387

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

multiStateOutput.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

multiStateOutput.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

multiStateOutput.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

multiStateOutput.numberOfStates

According to the BACnet protocol documentation:

Defines the number of states that the presentValue may have.

Property *numberOfStates* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.numberOfStates

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 74

Tag Value

A 32-bit unsigned integer.

multiStateOutput.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

multiStateOutput.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

multiStateOutput.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

multiStateOutput.outOfService

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateOutput.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *multiStateOutput* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit unsigned integer.

multiStateOutput.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *multiStateOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 14
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

1 elm01

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

2 elm02

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

3

elm03

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

4

elm04

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

5 elm05

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

6 elm06

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

7 elm07

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

8 elm08

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

9

elm09

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

10

elm10

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

11 elm11

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

12 elm12

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

13 elm13

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

14 elm14

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.

- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

15 elm15

One of the following data types:

- A null/empty value.
- A 32-bit floating point number.
- An unsigned integer representing an enumerated value.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<Boolean>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A character string containing an XML with a value of type **<Double>**.
- A Date/Time value (only the Time part).
- A string value.
- A character string containing an XML with a value of type **<OctetString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Date>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.

multiStateOutput.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

multiStateOutput.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

multiStateOutput.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

multiStateOutput.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

multiStateOutput.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall

have the value *NO_FAULT_DETECTED* unless *Out_Of_Service* is *TRUE* and an alternate value has been written to the *Reliability* property.

Property *reliabilityEvaluationInhibit* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateOutput.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A 32-bit unsigned integer.

multiStateOutput.stateText

According to the BACnet protocol documentation:

An array of character strings representing descriptions of all possible states of the presentValue.

Property *stateText* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.stateText

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 110

Tag Value

A character string containing an XML with a value of type **<SequenceOfCharacterString>**.

multiStateOutput.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *multiStateOutput* can be read and written using a block tag with 4 elements.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 14
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateOutput.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property tags of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

multiStateOutput.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

multiStateOutput.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

multiStateOutput.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *multiStateOutput* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 14
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

multiStateOutput.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *multiStateOutput* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateOutput-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 14
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types:

		• A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value.

- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

multiStateValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a multi-state value. A "multi-state value" is a control system parameter residing in the memory of the BACnet device. The actual functions associated with a specific state are a local matter and not specified by the protocol. For example, a particular state may represent the active/inactive condition of several physical inputs and outputs or perhaps the value of an analog input or output. The Present_Value property is an unsigned integer number representing the state. The State_Text property associates a description with each state.

Objects of type multiStateValue have the following properties:

- **ackedTransitions**
- **alarmValues**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultValues**
- **lastCommandTime**
- **notificationClass**
- **notifyType**
- **numberOfStates**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**

- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **stateText**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **valueSource**
- **valueSourceArray**

multiStateValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *multiStateValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 19
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateValue.alarmValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-OFFNORMAL event is generated.

Property *alarmValues* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.alarmValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 7

Tag Value

A character string containing an XML with a value of type **<SequenceOfUnsigned>**.

multiStateValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *multiStateValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 19
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

multiStateValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

multiStateValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

multiStateValue.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `multiStateValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateValue.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property `eventAlgorithmInhibitRef` of object `multiStateValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

multiStateValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *multiStateValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 19
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *multiStateValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 19
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

multiStateValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *multiStateValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 19
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

multiStateValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

multiStateValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *multiStateValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 19
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

multiStateValue.faultValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-FAULT event is generated.

Property *faultValues* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.faultValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 39

Tag Value

A character string containing an XML with a value of type **<SequenceOfUnsigned>**.

multiStateValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

multiStateValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

multiStateValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

multiStateValue.numberOfStates

According to the BACnet protocol documentation:

Defines the number of states that the presentValue may have.

Property *numberOfStates* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.numberOfStates

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 74

Tag Value

A 32-bit unsigned integer.

multiStateValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

multiStateValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

multiStateValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

multiStateValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *multiStateValue* can be read and written using a single IOTag.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit unsigned integer.

multiStateValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *multiStateValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 19
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
1	elm01	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
2	elm02	One of the following data types:

multiStateValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

multiStateValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

multiStateValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

multiStateValue.reliability

According to the BACnet protocol documentation:

Indicates whether the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

multiStateValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A 32-bit unsigned integer.

multiStateValue.stateText

According to the BACnet protocol documentation:

An array of character strings representing descriptions of all possible states of the presentValue.

Property *stateText* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.stateText

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 110

Tag Value

A character string containing an XML with a value of type **<SequenceOfCharacterString>**.

multiStateValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *multiStateValue* can be read and written using a block tag with 4 elements.

This property supports reading by COV (change of value notifications).

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 19
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

multiStateValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

multiStateValue.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

multiStateValue.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

multiStateValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *multiStateValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 19
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

multiStateValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *multiStateValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	multiStateValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 19
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

networkPort Object

According to the BACnet protocol documentation:

Provides access to the configuration and properties of network ports of a device. All BACnet devices shall contain at least one Network Port object per physical port which the device can be configured to communicate BACnet through (unless the port is currently for communications on a network other than the current BACnet internetwork and this use precludes its use for the current BACnet internetwork). It is a local matter whether or not Network Port objects exist for non-configured ports. It is a local matter whether or not the Network Port object is used for non-BACnet ports.

Objects of type networkPort have the following properties:

- **ackedTransitions**
- **apduLength**
- **autoSlaveDiscovery**
- **bacnetIpGlobalAddress**
- **bacnetIpMode**
- **bacnetIpMulticastAddress**
- **bacnetIpNatTraversal**
- **bacnetIpUdpPort**
- **bacnetIpv6Mode**
- **bacnetIpv6MulticastAddress**
- **bacnetIpv6UdpPort**
- **bbmdAcceptFdRegistrations**
- **bbmdBroadcastDistributionTable**
- **bbmdForeignDeviceTable**
- **changesPending**
- **command**
- **description**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **fdBbmdAddress**
- **fdSubscriptionLifetime**
- **ipAddress**

- **ipDefaultGateway**
- **ipDhcpEnable**
- **ipDhcpLeaseTime**
- **ipDhcpLeaseTimeRemaining**
- **ipDhcpServer**
- **ipDnsServer**
- **ipSubnetMask**
- **ipv6Address**
- **ipv6AutoAddressingEnable**
- **ipv6DefaultGateway**
- **ipv6DhcpLeaseTime**
- **ipv6DhcpLeaseTimeRemaining**
- **ipv6DhcpServer**
- **ipv6DnsServer**
- **ipv6PrefixLength**
- **ipv6ZoneIndex**
- **linkSpeed**
- **linkSpeedAutonegotiate**
- **linkSpeeds**
- **macAddress**
- **manualSlaveAddressBinding**
- **maxInfoFrames**
- **maxMaster**
- **networkInterfaceName**
- **networkNumber**
- **networkNumberQuality**
- **networkType**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**

- **outOfService**
- **profileLocation**
- **profileName**
- **propertyList**
- **protocolLevel**
- **referencePort**
- **reliability**
- **reliabilityEvaluationInhibit**
- **routingTable**
- **slaveAddressBinding**
- **slaveProxyEnable**
- **statusFlags**
- **tags**
- **virtualMacAddressTable**

networkPort.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *networkPort* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 56
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.apduLength

According to the BACnet protocol documentation:

Indicates the maximum number of octets that may be contained in a single indivisible application protocol data unit sent or received on this port. The value of this property shall be greater than or equal to 50. This property also indicates the maximum number of octets that may be contained in a single individual network service data unit sent or received on this port.

Property *apduLength* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.apduLength

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 399

Tag Value

A 32-bit unsigned integer.

networkPort.autoSlaveDiscovery

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the device will perform automatic slave detection functions for this port. This property shall be present if the device is capable of performing the functions of a Slave-Proxy device on this port.

Property *autoSlaveDiscovery* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.autoSlaveDiscovery

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 169

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.bacnetIpGlobalAddress

According to the BACnet protocol documentation:

Indicates the global address and UDP port from which the network port can be accessed from the global side of a NAT router. How the public IP address and UDP port are determined is a local matter. The 'none' choice in the BACnetHostAddress portion and a value of X'0000' in the port portion indicates that the global address cannot be determined or is not yet configured.

Property *bacnetIpGlobalAddress* of object *networkPort* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.bacnetIpGlobalAddress
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 56
B3:	<objectAddress>
B4:	<property> = 407
Size:	2 elements

Block Elements

Index	Name	Type
0	Host	One of the following data types: - A null/empty value. - A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A'). - A character string containing an XML with a value of type <CharacterString>.
1	Port	A 16-bit integer value ranging from 0 to 65535.

networkPort.bacnetIpMode

According to the BACnet protocol documentation:

Indicates the BACnet/IP mode of this network port.

Property *bacnetIpMode* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.bacnetIpMode

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 408

Tag Value

An integer representing one of the options of enumeration **<BACnetIPMode>**.

networkPort.bacnetIpMulticastAddress

According to the BACnet protocol documentation:

Contains the BACnet/IP multicast group address to be used for the distribution of BACnet broadcast messages. The value of this property shall be conveyed with the most significant octet first. A value of X'00000000' indicates that BACnet/IP multicast is not used.

Property *bacnetIpMulticastAddress* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.bacnetIpMulticastAddress

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 409

Tag Value

A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

networkPort.bacnetIpNatTraversal

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) this port is configured to operate in a NAT environment, and the global address is indicated by the value of the BACnet_IP_Global_Address property.

Property *bacnetIpNatTraversal* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.bacnetIpNatTraversal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 410

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.bacnetIpUdpPort

According to the BACnet protocol documentation:

Indicates the UDP port number of this network port.

Property *bacnetIpUdpPort* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.bacnetIpUdpPort

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 412

Tag Value

A 16-bit integer value ranging from 0 to 65535.

networkPort.bacnetIpv6Mode

According to the BACnet protocol documentation:

Indicates the BACnet/IPv6 mode of this network port.

Property *bacnetIpv6Mode* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.bacnetIpv6Mode

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 435

Tag Value

An integer representing one of the options of enumeration **<BACnetIPMode>**.

networkPort.bacnetIpv6MulticastAddress

According to the BACnet protocol documentation:

Contains the IPv6 multicast address UDP port to be used for the distribution of BACnet broadcast messages in the local multicast domain. The value of this property shall be comprised of the IPv6 multicast address followed by the UDP port, both of which shall be conveyed with the most significant octet first.

Property *bacnetIpv6MulticastAddress* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.bacnetIpv6MulticastAddress

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 440

Tag Value

A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

networkPort.bacnetIpv6UdpPort

According to the BACnet protocol documentation:

Indicates the BACnet/IPv6 UDP port number of this network port.

Property *bacnetIpv6UdpPort* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.bacnetIpv6UdpPort

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 438

Tag Value

A 16-bit integer value ranging from 0 to 65535.

networkPort.bbmdAcceptFdRegistrations

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) this device shall accept foreign device registrations.

Property *bbmdAcceptFdRegistrations* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.bbmdAcceptFdRegistrations

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 413

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.bbmdBroadcastDistributionTable

According to the BACnet protocol documentation:

Reflects the current value of the broadcast distribution table.

Property *bbmdBroadcastDistributionTable* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.bbmdBroadcastDistributionTable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 414

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetBDTEntry>**.

networkPort.bbmdForeignDeviceTable

According to the BACnet protocol documentation:

Reflects the current value of the foreign device table.

Property *bbmdForeignDeviceTable* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.bbmdForeignDeviceTable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 415

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetFDTEEntry>**.

networkPort.changesPending

According to the BACnet protocol documentation:

Indicates whether the configuration settings in the Network Port object map to the current configuration settings. A value of FALSE indicates that the configuration settings reflect the current port configuration information. A value of TRUE indicates the configuration settings have been modified but have not been activated on the port. When a value is written to a property that requires activation, the value of the Changes_Pending property shall automatically be set to TRUE, indicating that the current property values are not the values actively in use.

Property *changesPending* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.changesPending

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 416

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.command

According to the BACnet protocol documentation:

Used to request that the Network Port object perform various actions.

Property *command* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.command

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 417

Tag Value

An integer representing one of the options of enumeration **<BACnetNetworkPortCommand>**.

networkPort.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

networkPort.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *networkPort* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 56
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *networkPort* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 56
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

networkPort.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *networkPort* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 56
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

networkPort.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

networkPort.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *networkPort* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 56
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

networkPort.fdBbmdAddress

According to the BACnet protocol documentation:

Indicates the BBMD with which the local device is to register as a foreign device when BACnet_IP_Mode is FOREIGN. This property shall be present and writable if BACnet_IP_Mode is FOREIGN.

Property *fdBbmdAddress* of object *networkPort* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.fdBbmdAddress
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 56
B3:	<objectAddress>
B4:	<property> = 418
Size:	2 elements

Block Elements

Index	Name	Type
0	Host	One of the following data types: • A null/empty value. • A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A'). • A character string containing an XML with a value of type <CharacterString>.
1	Port	A 16-bit integer value ranging from 0 to 65535.

networkPort.fdSubscriptionLifetime

According to the BACnet protocol documentation:

Indicates the Time-To-Live value, in seconds, to be used in the Register-Foreign-Device BVLL message. This property shall be present and writable if BACnet_IP_Mode is FOREIGN.

Property *fdSubscriptionLifetime* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.fdSubscriptionLifetime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 419

Tag Value

A 16-bit integer value ranging from 0 to 65535.

networkPort.ipAddress

According to the BACnet protocol documentation:

Indicates the nominal power consumption of the load(s) controlled by this object at this moment. The units shall be kilowatts.

Property *ipAddress* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipAddress

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 400

Tag Value

A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

networkPort.ipDefaultGateway

According to the BACnet protocol documentation:

Indicates the IP address of the default gateway for this network. This property shall be conveyed with the most significant octet first. If the IP_DHCP_Enable property is TRUE, this property shall be read-only.

Property *ipDefaultGateway* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipDefaultGateway

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 401

Tag Value

A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

networkPort.ipDhcpEnable

According to the BACnet protocol documentation:

Indicates whether or not this network is configured via DHCP. A value of TRUE indicates that DHCP configuration is enabled, FALSE indicates it is not. This property is required if DHCP configuration is supported by this network port.

Property *ipDhcpEnable* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipDhcpEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 402

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.ipDhcpLeaseTime

According to the BACnet protocol documentation:

Indicates the lease time in seconds of the last DHCP lease obtained for the port. If IP_DHCP_Enable is FALSE, or no lease has been acquired, or the value is unknown, this property shall be 0.

Property *ipDhcpLeaseTime* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipDhcpLeaseTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 403

Tag Value

A 32-bit unsigned integer.

networkPort.ipDhcpLeaseTimeRemaining

According to the BACnet protocol documentation:

Indicates the lease time in seconds remaining of the last DHCP lease obtained for the port. If IP_DHCP_Enable is FALSE, or no lease has been acquired, or the value is unknown, this property shall be 0.

Property *ipDhcpLeaseTimeRemaining* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipDhcpLeaseTimeRemaining

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 404

Tag Value

A 32-bit unsigned integer.

networkPort.ipDhcpServer

According to the BACnet protocol documentation:

Indicates the address of the DHCP server from which the last DHCP lease was obtained for the port. If the address of the DHCP server cannot be determined, the value of this property shall be X'00000000'.

Property *ipDhcpServer* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipDhcpServer

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 405

Tag Value

A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

networkPort.ipDnsServer

According to the BACnet protocol documentation:

Indicates the DNS server used by this network port for Internet host name resolution. The values of this property shall be conveyed with the most significant octet first. A value of X'00000000' in an array entry indicates that the DNS server address is not available or is not configured.

Property *ipDnsServer* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipDnsServer

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 406

Tag Value

A character string containing an XML with a value of type **<SequenceOfOctetString>**.

networkPort.ipSubnetMask

According to the BACnet protocol documentation:

Indicates the subnet mask for this network. This property shall be conveyed with the most significant octet first. If the IP_DHCP_Enable property is TRUE, this property shall be read-only.

Property *ipSubnetMask* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipSubnetMask

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 411

Tag Value

A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

networkPort.ipv6Address

According to the BACnet protocol documentation:

Indicates the IPv6 address of this network port. This property shall be conveyed most significant octet first. If the IPv6 address is obtained automatically, this property shall be read-only, and the value of this property shall contain the address with the highest precedence.

Property *ipv6Address* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipv6Address

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 436

Tag Value

A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

networkPort.ipv6AutoAddressingEnable

According to the BACnet protocol documentation:

Indicates whether or not this network is configured for automatic address assignment via DHCPv6, Stateless Auto Address Configuration (SLAAC, RFC 4862), or neighbor discovery. A value of TRUE indicates that automatic address assignment is enabled, FALSE indicates it is not.

Property *ipv6AutoAddressingEnable* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipv6AutoAddressingEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 442

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.ipv6DefaultGateway

According to the BACnet protocol documentation:

Indicates the IPv6 address of the default gateway for this network. This property shall be conveyed with the most significant octet first. If the IPv6 address is obtained automatically, this property shall be read-only.

Property *ipv6DefaultGateway* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipv6DefaultGateway

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 439

Tag Value

A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

networkPort.ipv6DhcpLeaseTime

According to the BACnet protocol documentation:

Indicates the lease time in seconds of the last DHCPv6 lease obtained for the port. If IPv6_Auto_Addressing_Enable is FALSE, or DHCPv6 is not in use, or no lease has been acquired, or the value is unknown, this property shall be 0.

Property *ipv6DhcpLeaseTime* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipv6DhcpLeaseTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 443

Tag Value

A 32-bit unsigned integer.

networkPort.ipv6DhcpLeaseTimeRemaining

According to the BACnet protocol documentation:

Indicates the lease time in seconds remaining of the last DHCPv6 lease obtained for the port. If IPv6_Auto_Addressing_Enable is FALSE, or DHCPv6 is not in use, or no lease has been acquired, or the value is unknown, this property shall be 0.

Property *ipv6DhcpLeaseTimeRemaining* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipv6DhcpLeaseTimeRemaining

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 444

Tag Value

A 32-bit unsigned integer.

networkPort.ipv6DhcpServer

According to the BACnet protocol documentation:

Indicates the address of the DHCPv6 server from which the last DHCPv6 lease was obtained for the port. If the address of the DHCPv6 server cannot be determined, or DHCPv6 is not in use, the value of this property shall be X'00000000000000000000000000000000'.

Property *ipv6DhcpServer* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipv6DhcpServer

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 445

Tag Value

A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

networkPort.ipv6DnsServer

According to the BACnet protocol documentation:

Indicates the IPv6 address of the DNS server used by this network port for Internet host name resolution. The values of this property shall be conveyed with the most significant octet first. If the DNS server addresses are obtained automatically, this property shall be read-only. A value of X'00000000000000000000000000000000' in an array entry indicates that the DNS server address is not available or is not configured.

Property *ipv6DnsServer* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipv6DnsServer

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 441

Tag Value

A character string containing an XML with a value of type **<SequenceOfOctetString>**.

networkPort.ipv6PrefixLength

According to the BACnet protocol documentation:

Indicates the length in bits of the subnet prefix of the IPv6 address of this network port. The value of this property shall be in the range 1 to 128. If the IPv6 address is obtained automatically, this property shall be read-only.

Property *ipv6PrefixLength* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipv6PrefixLength

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 437

Tag Value

An 8-bit integer value ranging from 0 to 255.

networkPort.ipv6ZoneIndex

According to the BACnet protocol documentation:

Contains the zone index for the B/IPv6 link local address when the node supports multiple IPv6 link local addresses. According to RFC 4007, because all link-local addresses in a host have a common prefix, normal routing procedures cannot be used to choose the outgoing interface when sending packets to a link-local destination. A special identifier, known as a zone index, is needed to provide the additional routing information.

Property *ipv6ZoneIndex* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.ipv6ZoneIndex

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 446

Tag Value

A string value.

networkPort.linkSpeed

According to the BACnet protocol documentation:

Represents the network communication rate as the number of bits per second. A value of 0 indicates an unknown communication rate.

Property *linkSpeed* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.linkSpeed

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 420

Tag Value

A 32-bit floating point number.

networkPort.linkSpeedAutonegotiate

According to the BACnet protocol documentation:

Represents the auto negotiation setting of the network port. A value of TRUE indicates that the device automatically determines the speed of this network port. In this case, Link_Speed shall be read-only and indicate the determined speed, if available. A value of FALSE indicates that the link speed is determined by the value of the Link_Speed property.

Property *linkSpeedAutonegotiate* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.linkSpeedAutonegotiate

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 422

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.linkSpeeds

According to the BACnet protocol documentation:

Contains an array of the link speeds supported by this network port.

Property *linkSpeeds* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.linkSpeeds

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 421

Tag Value

A character string containing an XML with a value of type **<SequenceOfREAL>**.

networkPort.macAddress

According to the BACnet protocol documentation:

Contains the BACnet MAC address used on this network. The value of this property shall be conveyed with the most significant octet first. If Network_Type is IPV4 and the Protocol_Level is BACNET_APPLICATION, then the value of this property shall contain the six octet combination of the IP_Address and BACnet_IP_UDP_Port and shall be read-only. If the value of Network_Type is a value that represents a port that requires VMAC addressing, then the value of this property shall be read-only and contain the VMAC address.

Property *macAddress* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.macAddress

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 423

Tag Value

A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

networkPort.manualSlaveAddressBinding

According to the BACnet protocol documentation:

Describes the manually configured set of slave devices for which this device is acting as a Slave Proxy. This property shall be present and writable if the device is capable of performing the functions of a Slave-Proxy device on this port.

Property *manualSlaveAddressBinding* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.manualSlaveAddressBinding

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 170

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAddressBinding>**.

networkPort.maxInfoFrames

According to the BACnet protocol documentation:

Specifies the maximum number of information frames the node may send before it must pass the token.

Property *maxInfoFrames* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.maxInfoFrames

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 63

Tag Value

An 8-bit integer value ranging from 0 to 255.

networkPort.maxMaster

According to the BACnet protocol documentation:

Specifies the highest possible address for master nodes, and shall be less than or equal to 127.

Property *maxMaster* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.maxMaster

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 64

Tag Value

An 8-bit integer value ranging from 0 to 255.

networkPort.networkInterfaceName

According to the BACnet protocol documentation:

Identifies the network interface hardware to which this network port is bound. For example, if Network_Type is IPV4, the value of this property identifies the Ethernet hardware interface that this network port is using to communicate.

Property *networkInterfaceName* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.networkInterfaceName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 424

Tag Value

A string value.

networkPort.networkNumber

According to the BACnet protocol documentation:

Represents the BACnet network number associated with this network. The range for this property shall be 0 .. 65534. A Network_Number of 0 indicates that the Network_Number is not known or cannot be determined. Writing 0 to the Network_Number property shall force the value of the Network_Number_Quality property to UNKNOWN and allows the device to attempt to learn the network number, if possible. Writing a value other than 0 shall force the Network_Number_Quality property to CONFIGURED. If the Network_Type is PTP, then this property shall be read-only and contain a value of 0.

Property *networkNumber* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.networkNumber

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 425

Tag Value

A 16-bit integer value ranging from 0 to 65535.

networkPort.networkNumberQuality

According to the BACnet protocol documentation:

Represents the current quality of the Network_Number property. If the Network_Type is PTP, the Network_Number_Quality shall be CONFIGURED.

Property *networkNumberQuality* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.networkNumberQuality

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 426

Tag Value

An integer representing one of the options of enumeration **<BACnetNetworkNumberQuality>**.

networkPort.networkType

According to the BACnet protocol documentation:

Represents the type of network this Network Port object is representing. When the Protocol_Level is BACNET_APPLICATION, the Network_Type indicates the protocol over which BACnet is operating and implies that the requirements laid out in the appropriate clause are being met. For example, if the Network_Type is IPV4, then the port is operating as a BACnet/IP port.

Property *networkType* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.networkType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 427

Tag Value

An integer representing one of the options of enumeration **<BACnetNetworkType>**.

networkPort.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

networkPort.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

networkPort.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

networkPort.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

networkPort.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

networkPort.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

networkPort.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

networkPort.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

networkPort.protocolLevel

According to the BACnet protocol documentation:

Indicates whether the object represents a physical network interface (PHYSICAL), a non-BACnet protocol (PROTOCOL), the BACnet use of the protocol (BACNET_APPLICATION), or a non-BACnet use of the protocol (NON_BACNET_APPLICATION).

Property *protocolLevel* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.protocolLevel

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 482

Tag Value

An integer representing one of the options of enumeration **<BACnetProtocolLevel>**.

networkPort.referencePort

According to the BACnet protocol documentation:

Specifies the instance of the Network Port object that this Network Port object uses as its lower protocol layer (i.e. transport, routing, datalink, etc). This property allows the Network Port objects in the device to describe the hierarchy of protocols and physical ports in order to support complex network configuration required by some advanced BACnet products. If this property is absent and the Protocol_Level is BACNET_APPLICATION, then it represents all protocol layers in a single object. If this property has a value of 4194303, then this object has not been assigned a lower protocol layer. If the object is capable of representing all protocol layers in a single object, then this is a valid configuration and the object shall behave as if this property were absent. If the object is not capable of representing all protocol layers in a single object, then this is an indication that the object is not yet configured. A Network Port object is misconfigured if the referenced Network Port object has a Protocol_Level of BACNET_APPLICATION, or the referenced Network Port object does not exist.

Property *referencePort* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.referencePort

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 483

Tag Value

A 32-bit unsigned integer.

networkPort.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

networkPort.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.routingTable

According to the BACnet protocol documentation:

Contains the table of first hop routers to remote networks reachable through this port.

Property *routingTable* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.routingTable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 428

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetRouterEntry>**.

networkPort.slaveAddressBinding

According to the BACnet protocol documentation:

Describes the set of slave devices for which this device is acting as a Slave-Proxy on this port. This property shall be present if the device is capable of performing the functions of a Slave-Proxy device on this port. The set of devices described by the Slave_Address_Binding property consists of those devices described in the Manual_Slave_Address_Binding and those devices that are automatically discovered. When enabled, the Slave-Proxy device shall periodically check each device in this list by reading the device's Protocol_Services_Supported property from the device's Device object using the ReadProperty service. If the device fails to respond or indicates that it executes the Who-Is service, it shall be removed from the list. The period at which the devices are checked is a local matter.

Property *slaveAddressBinding* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.slaveAddressBinding

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 171

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetAddressBinding>**.

networkPort.slaveProxyEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the device will perform Slave-Proxy functions for this port. This property shall be present and writable if the device is capable of performing the functions of a Slave-Proxy device on this port.

Property *slaveProxyEnable* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.slaveProxyEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 172

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *networkPort* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 56
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkPort.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property tags of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

networkPort.virtualMacAddressTable

According to the BACnet protocol documentation:

Contains the list of VMAC entries.

Property *virtualMacAddressTable* of object *networkPort* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkPort-<objectAddress>.virtualMacAddressTable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 56
N3:	<objectAddress>
N4:	<property> = 429

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetVMACEntry>**.

networkSecurity Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible network security settings and status of a BACnet device. Secure BACnet devices shall contain exactly one Network Security object and they shall have an instance of 1. Operations on the Network Security object shall always be deemed to have sufficient authorization if the request is secured with an Installation key.

Objects of type networkSecurity have the following properties:

- **baseDeviceSecurityPolicy**
- **description**
- **distributionKeyRevision**
- **doNotHide**
- **keySets**
- **lastKeyServer**
- **networkAccessSecurityPolicies**
- **objectIdentifier**
- **objectName**
- **objectType**
- **packetReorderTime**
- **profileLocation**
- **profileName**
- **propertyList**

- **securityPduTimeout**
- **securityTimeWindow**
- **supportedSecurityAlgorithms**
- **tags**
- **updateKeySetTimeout**

networkSecurity.baseDeviceSecurityPolicy

According to the BACnet protocol documentation:

Specifies the minimum level of security that the device requires allowing client devices to know the level of security to use when communicating with the device. While devices may require higher security levels for some operations, this property shall be readable using the security level defined by this property.

Property *baseDeviceSecurityPolicy* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.baseDeviceSecurityPolicy

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 327

Tag Value

An integer representing one of the options of enumeration **<BACnetSecurityLevel>**.

networkSecurity.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

networkSecurity.distributionKeyRevision

According to the BACnet protocol documentation:

Identifies the device's Distribution key revision. This property shall be 0 if the device does not have a Distribution key.

Property *distributionKeyRevision* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.distributionKeyRevision

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 328

Tag Value

An 8-bit integer value ranging from 0 to 255.

networkSecurity.doNotHide

According to the BACnet protocol documentation:

Indicates whether or not the device is allowed to ignore certain network security error conditions.

Property *doNotHide* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.doNotHide

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 329

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

networkSecurity.keySets

According to the BACnet protocol documentation:

Describes the contents of the device's 2 key sets. The actual key values are not included in the contents of this property. When a key set has not been provided, the keyrevision field shall be set to 0, the key-ids field shall be empty, and the activation-time and expiration-time fields shall contain all wildcard values.

Property *keySets* of object *networkSecurity* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.keySets
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 38
B3:	<objectAddress>
B4:	<property> = 330
Size:	2 elements

Block Elements

Index	Name	Type
0	elm00	A character string containing an XML with a value of type <BACnetSecurityKeySet> .
1	elm01	A character string containing an XML with a value of type <BACnetSecurityKeySet> .

networkSecurity.lastKeyServer

According to the BACnet protocol documentation:

Specifies the device identifier and address of the last Key Server that successfully updated a security key in the device. If no Key Server has updated the keys sets in the device, the deviceidentifier field shall contain 4194303 in the instance part, the network-number field shall be 0, and the mac-address field shall be empty. This property is writable in order to allow a Key Server address to be provided to the secure device before it has received a Device-Master key. This allows the secure device

to be directed to the Key Server in a legacy environment where globally broadcast Request-Key-Update messages will not be routed. A device may make this property read-only once a Device-Master key has been received.

Property `lastKeyServer` of object `networkSecurity` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.lastKeyServer
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 38
B3:	<objectAddress>
B4:	<property> = 331
Size:	3 elements

Block Elements

Index	Name	Type
0	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>--<Address>' (ex: 'analogOutput-345').
1	DeviceAddress_Network Number	A 16-bit integer value ranging from 0 to 65535.
2	DeviceAddress_MacAddress	A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

networkSecurity.networkAccessSecurityPolicies

According to the BACnet protocol documentation:

Specifies the security policy for each network directly connected to the device. It specifies the level of security that the device should use for network infrastructure services, such as Who-Is, I-Am, Who-Is-Router, etc. This array shall have 1 entry for each network port. The Port ID field shall correspond to the Port ID of the associated network. For non-routing nodes, this value shall be 0.

Property `networkAccessSecurityPolicies` of object `networkSecurity` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.networkAccessSecurityPolicies

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 332

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNetworkSecurityPolicy>**.

networkSecurity.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

networkSecurity.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

networkSecurity.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

networkSecurity.packetReorderTime

According to the BACnet protocol documentation:

Specifies the packet reorder time, in milliseconds, used by the device for validating Message Ids. The recommended default value for this property is 500 (0.5 seconds). The property shall be restricted to the range 1 through 3000.

Property *packetReorderTime* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.packetReorderTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 333

Tag Value

A 32-bit unsigned integer.

networkSecurity.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

networkSecurity.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

networkSecurity.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

networkSecurity.securityPduTimeout

According to the BACnet protocol documentation:

Specifies the length of time, in milliseconds, the device waits for a security response. For the application TSM to work correctly, this value should be configured to be less than the APDU_Segment_Timeout value in the Device object.

Property *securityPduTimeout* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.securityPduTimeout

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 334

Tag Value

A 16-bit integer value ranging from 0 to 65535.

networkSecurity.securityTimeWindow

According to the BACnet protocol documentation:

Specifies the security time window for the device in seconds. The recommended default value for this property is 180 (3 minutes). The property shall be restricted to the range 1 through 600.

Property *securityTimeWindow* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.securityTimeWindow

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 335

Tag Value

A 32-bit unsigned integer.

networkSecurity.supportedSecurityAlgorithms

According to the BACnet protocol documentation:

Identifies the encryption and signature algorithm pairs that the device supports.

Property *supportedSecurityAlgorithms* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.supportedSecurityAlgorithms

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 336

Tag Value

A character string containing an XML with a value of type **<SequenceOfUnsigned8>**.

networkSecurity.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *networkSecurity* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

networkSecurity.updateKeySetTimeout

According to the BACnet protocol documentation:

Indicates the maximum amount of time, in milliseconds, that the device will take to respond to an Update-Key-Set message. This value added to the device APDU_Timeout results in the amount of time that a Key Server shall wait for a Security-Response for an Update-Key-Set message. The use of APDU_Timeout is to allow for network delay; whereas the Update_Key_Set_Timeout provides for the actual time the device will need to apply the keys to its key set.

Property `updateKeySetTimeout` of object `networkSecurity` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	networkSecurity-<objectAddress>.updateKeySetTimeout

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 38
N3:	<objectAddress>
N4:	<property> = 337

Tag Value

A 16-bit integer value ranging from 0 to 65535.

notificationClass Object

According to the BACnet protocol documentation:

Defines a standardized object that represents and contains information required for the distribution of event notifications within BACnet systems. Notification Classes are useful for event-initiating objects that have identical needs in terms of how their notifications should be handled, what the destination(s) for their notifications should be, and how they should be acknowledged.

Objects of type `notificationClass` have the following properties:

- `ackRequired`
- `ackedTransitions`
- `description`
- `eventDetectionEnable`
- `eventEnable`
- `eventMessageTexts`
- `eventMessageTextsConfig`
- `eventState`
- `eventTimeStamps`
- `notificationClass`
- `notifyType`

- **objectIdentifier**
- **objectName**
- **objectType**
- **priority**
- **profileLocation**
- **profileName**
- **propertyList**
- **recipientList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**
- **tags**

notificationClass.ackRequired

According to the BACnet protocol documentation:

Conveys three separate flags that represent whether acknowledgement shall be required in notifications generated for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL event transitions, respectively.

Property *ackRequired* of object *notificationClass* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.ackRequired
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 15
B3:	<objectAddress>
B4:	<property> = 1
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

notificationClass.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *notificationClass* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 15
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

notificationClass.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

notificationClass.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

notificationClass.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property `eventEnable` of object `notificationClass` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 15
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

notificationClass.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property `eventMessageTexts` of object `notificationClass` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 15
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

notificationClass.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *notificationClass* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 15
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

notificationClass.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

notificationClass.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *notificationClass* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 15
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

notificationClass.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

notificationClass.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

notificationClass.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

notificationClass.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

notificationClass.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

notificationClass.priority

According to the BACnet protocol documentation:

Conveys the priority to be used when issuing event notifications in the case when a Notification Class object is not used.

Property *priority* of object *notificationClass* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.priority
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 15
B3:	<objectAddress>
B4:	<property> = 86
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A 32-bit unsigned integer.
1	elm01	A 32-bit unsigned integer.
2	elm02	A 32-bit unsigned integer.

notificationClass.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

notificationClass.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

notificationClass.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

notificationClass.recipientList

According to the BACnet protocol documentation:

Conveys a list of one or more recipient destinations to which notifications shall be sent when event initiating objects using this class detect the occurrence of an event.

Property *recipientList* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.recipientList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 102

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDestination>**.

notificationClass.reliability

According to the BACnet protocol documentation:

Indicates whether the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

notificationClass.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

notificationClass.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *notificationClass* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 15
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

notificationClass.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *notificationClass* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationClass-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 15
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

notificationForwarder Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics required for the re-distribution of event notifications to zero or more destinations. It differs from a Notification Class in that the Notification Forwarder object is not used for originating event notifications, but rather is used to forward event notifications to a different and potentially larger number of recipients.

Objects of type notificationForwarder have the following properties:

- **description**
- **localForwardingOnly**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **portFilter**
- **processIdentifierFilter**
- **profileLocation**
- **profileName**
- **propertyList**
- **recipientList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**
- **subscribedRecipients**
- **tags**

notificationForwarder.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

notificationForwarder.localForwardingOnly

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the object is limited to forwarding notifications initiated from within the same device. If Local_Forwarding_Only has a value of FALSE, then the Notification Forwarder is capable of forwarding notifications for other devices.

Property *localForwardingOnly* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.localForwardingOnly

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 360

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

notificationForwarder.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

notificationForwarder.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

notificationForwarder.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

notificationForwarder.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

notificationForwarder.portFilter

According to the BACnet protocol documentation:

Enables or disables the forwarding of event notifications received on a particular network port. When an event notification is received on a port that is marked as disabled by this property, the Notification Forwarder object shall ignore that event notification.

Property *portFilter* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.portFilter

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 363

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPortPermission>**.

notificationForwarder.processIdentifierFilter

According to the BACnet protocol documentation:

Used in the selection of event notification service requests that are to be forwarded by the object. When the Process Identifier parameter of a received event notification is the same as the value of the Process_Identifier_Filter property, or if the Process_Identifier_Filter property contains a NULL, then the notification will be accepted for forwarding by the Notification Forwarder object subject to the port, network and broadcast restrictions.

Property *processIdentifierFilter* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.processIdentifierFilter

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 361

Tag Value

One of the following data types:

- A 32-bit unsigned integer.
- A null/empty value.

notificationForwarder.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

notificationForwarder.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

notificationForwarder.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

notificationForwarder.recipientList

According to the BACnet protocol documentation:

Conveys a list of one or more recipient destinations to which notifications shall be sent when event initiating objects using this class detect the occurrence of an event.

Property *recipientList* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.recipientList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 102

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDestination>**.

notificationForwarder.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

notificationForwarder.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall

have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

notificationForwarder.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *notificationForwarder* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 51
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

notificationForwarder.subscribedRecipients

According to the BACnet protocol documentation:

Conveys a list of recipient destinations to which event notifications are sent when events are forwarded by the Notification Forwarder object. These recipient destinations are intended to be temporary, and will expire if not renewed.

Property *subscribedRecipients* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.subscribedRecipients

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 362

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetEventNotificationSubscription>**.

notificationForwarder.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *notificationForwarder* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	notificationForwarder-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 51
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

octetstringValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use an OctetString Value object to make any kind of OCTET STRING data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

Objects of type octetstringValue have the following properties:

- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **eventState**
- **lastCommandTime**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **statusFlags**
- **tags**
- **valueSource**

- `valueSourceArray`

`octetstringValue.commandTimeArray`

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property `commandTimeArray` of object `octetstringValue` can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 47
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

		• A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
7	elm07	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
8	elm08	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
9	elm09	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set;

		• A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
10	elm10	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
11	elm11	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
12	elm12	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
13	elm13	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
14	elm14	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
15	elm15	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

octetstringValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

octetstringValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

octetstringValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

octetstringValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

octetstringValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

octetstringValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

octetstringValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

octetstringValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

octetstringValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

octetstringValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *octetstringValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 47
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
1	elm01	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
2	elm02	One of the following data types:

octetstringValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

octetstringValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

octetstringValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

octetstringValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

octetstringValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

octetstringValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').

octetstringValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *octetstringValue* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 47
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

octetstringValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

octetstringValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *octetstringValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 47
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

octetstringValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *octetstringValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	octetstringValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 47
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

positiveIntegerValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a Positive Integer Value object to make any kind of unsigned data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

Objects of type `positiveIntegerValue` have the following properties:

- **ackedTransitions**
- **commandTimeArray**
- **covIncrement**
- **currentCommandPriority**
- **deadband**
- **description**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **faultHighLimit**
- **faultLowLimit**
- **highLimit**
- **lastCommandTime**
- **limitEnable**
- **lowLimit**
- **maxPresValue**
- **minPresValue**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**

- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **resolution**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **units**
- **valueSource**
- **valueSourceArray**

positiveIntegerValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *positiveIntegerValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 48
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

positiveIntegerValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *positiveIntegerValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 48
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

		• A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
7	elm07	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
8	elm08	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

		• A character string containing an XML with a value of type <BACnetTimeStamp>.
9	elm09	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
10	elm10	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
11	elm11	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
12	elm12	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
13	elm13	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
14	elm14	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
15	elm15	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set;

- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

positiveIntegerValue.covIncrement

According to the BACnet protocol documentation:

Specifies the minimum change in presentValue that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

Property *covIncrement* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.covIncrement

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 22

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

positiveIntegerValue.deadband

According to the BACnet protocol documentation:

Specifies a range between the highLimit and lowLimit properties, which the presentValue must remain within for a TO-NORMAL event to be generated.

Property *deadband* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.deadband

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 25

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

positiveIntegerValue.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property `eventAlgorithmInhibit` of object `positiveIntegerValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

positiveIntegerValue.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property `Event_Algorithm_Inhibit`. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type `BACnetBinaryPV` or `BOOLEAN`.

Property `eventAlgorithmInhibitRef` of object `positiveIntegerValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '<index>)' can be used on array properties.

positiveIntegerValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, `Event_State` shall be NORMAL, and the properties `Acked_Transitions`, `Event_Time_Stamps`, and `Event_Message_Texts` shall be equal to their respective initial conditions.

Property `eventDetectionEnable` of object `positiveIntegerValue` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

positiveIntegerValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *positiveIntegerValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 48
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

positiveIntegerValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *positiveIntegerValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 48
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

positiveIntegerValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *positiveIntegerValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 48
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

positiveIntegerValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

positiveIntegerValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *positiveIntegerValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 48
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

positiveIntegerValue.faultHighLimit

According to the BACnet protocol documentation:

Specifies a limit that the Present_Value must exceed before a fault event is generated by the FAULT_OUT_OF_RANGE fault algorithm.

Property *faultHighLimit* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.faultHighLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 388

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.faultLowLimit

According to the BACnet protocol documentation:

Specifies a limit that the Present_Value must fall below before a fault event is generated by the FAULT_OUT_OF_RANGE fault algorithm.

Property *faultLowLimit* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.faultLowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 389

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.highLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must exceed before an event is generated.

Property *highLimit* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.highLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 45

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

positiveIntegerValue.limitEnable

According to the BACnet protocol documentation:

Conveys two flags that separately enable and disable reporting of highLimit and lowLimit offnormal events and their return to normal.

Property *limitEnable* of object *positiveIntegerValue* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.limitEnable
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 48
B3:	<objectAddress>
B4:	<property> = 52
Size:	2 elements

Block Elements

Index	Name	Type
0	lowLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	highLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

positiveIntegerValue.lowLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must fall below before an event is generated.

Property *lowLimit* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.lowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 59

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.maxPresValue

According to the BACnet protocol documentation:

Indicates the highest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *maxPresValue* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.maxPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 65

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.minPresValue

According to the BACnet protocol documentation:

Indicates the lowest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *minPresValue* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.minPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 69

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

positiveIntegerValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

positiveIntegerValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

positiveIntegerValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

positiveIntegerValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

positiveIntegerValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *positiveIntegerValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 48
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
1	elm01	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
2	elm02	One of the following data types:

positiveIntegerValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

positiveIntegerValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

positiveIntegerValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

positiveIntegerValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

positiveIntegerValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

positiveIntegerValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.resolution

According to the BACnet protocol documentation:

Indicates the smallest recognizable change in presentValue in engineering units (read-only).

Property *resolution* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.resolution

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 106

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *positiveIntegerValue* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 48
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

positiveIntegerValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

positiveIntegerValue.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

positiveIntegerValue.units

According to the BACnet protocol documentation:

Indicates the measurement units of this object.

Property *units* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.units

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 117

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

positiveIntegerValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *positiveIntegerValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 48
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

positiveIntegerValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *positiveIntegerValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	positiveIntegerValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 48
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

program Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of an application program. In this context, an application program is an abstract representation of a process within a BACnet device, which is executing a particular body of instructions that act upon a particular collection of data structures. The logic that is embodied in these instructions and the form and content of these data structures are local matters.

Objects of type program have the following properties:

- **ackedTransitions**
- **description**
- **descriptionOfHalt**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **instanceOf**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **profileLocation**
- **profileName**
- **programChange**
- **programLocation**
- **programState**
- **propertyList**
- **reasonForHalt**
- **reliability**
- **reliabilityEvaluationInhibit**
- **statusFlags**
- **tags**

program.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *program* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 16
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

program.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

program.descriptionOfHalt

According to the BACnet protocol documentation:

String used to describe the reason why a program has been halted.

Property *descriptionOfHalt* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.descriptionOfHalt

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 29

Tag Value

A string value.

program.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

program.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *program* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 16
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

program.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *program* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 16
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

program.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *program* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 16
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

program.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

program.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *program* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 16
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

program.instanceOf

According to the BACnet protocol documentation:

The local name of the application program being executed by this process.

Property *instanceOf* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.instanceOf

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 48

Tag Value

A string value.

program.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

program.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

program.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

program.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

program.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

program.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

program.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

program.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

program.programChange

According to the BACnet protocol documentation:

Used to request changes to the operating state of the process this object represents. 0 = ready, 1 = load, 2 = run, 3 = halt, 4 = restart, 5 = unload

Property *programChange* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.programChange

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 90

Tag Value

An integer representing one of the options of enumeration **<BACnetProgramRequest>**.

program.programLocation

According to the BACnet protocol documentation:

String that may be used by the application program to indicate its location within the program code, for example, a line number or program label or section name.

Property *programLocation* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.programLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 91

Tag Value

A string value.

program.programState

According to the BACnet protocol documentation:

Reflects the current logical state of the process executing the application program this object represents. 0 = idle, 1 = loading, 2 = running, 3 = waiting, 4 = halted, 5 = unloading

Property *programState* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.programState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 92

Tag Value

An integer representing one of the options of enumeration **<BACnetProgramState>**.

program.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

program.reasonForHalt

According to the BACnet protocol documentation:

Reflects the reason why the process was halted. 0 = normal, 1 = load failed, 2 = internal, 3 = program, 4 = other

Property *reasonForHalt* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.reasonForHalt

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 100

Tag Value

An integer representing one of the options of enumeration **<BACnetProgramError>**.

program.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

program.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

program.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *program* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 16
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

program.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *program* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	program-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 16
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

pulseConverter Object

According to the BACnet protocol documentation:

Defines a standardized object that represents a process whereby ongoing measurements made of some quantity, such as electric power or water or natural gas usage, and represented by pulses or counts, might be monitored over some time interval for applications such as peak load management, where it is necessary to make periodic measurements but where a precise accounting of every input pulse or count is not required.

Objects of type pulseConverter have the following properties:

- **ackedTransitions**
- **adjustValue**
- **count**
- **countBeforeChange**
- **countChangeTime**
- **covIncrement**
- **covPeriod**
- **deadband**
- **description**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**

- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **highLimit**
- **inputReference**
- **limitEnable**
- **lowLimit**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **scaleFactor**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **units**
- **updateTime**

pulseConverter.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *pulseConverter* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 24
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

pulseConverter.adjustValue

According to the BACnet protocol documentation:

Adjusts the Occupancy_Count property when written. When this property is written and the value of the Occupancy_Count_Enable property is FALSE, then the Adjust_Value property shall be set to zero. If Adjust_Value has never been written or the Occupancy_Count_Enable property is FALSE, then this property shall have a value of zero.

Property *adjustValue* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.adjustValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 176

Tag Value

A 32-bit floating point number.

pulseConverter.count

According to the BACnet protocol documentation:

Indicates the count of the input pulses as acquired from the physical input or the property referenced by the Input_Reference property.

Property *count* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.count

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 177

Tag Value

A 32-bit unsigned integer.

pulseConverter.countBeforeChange

According to the BACnet protocol documentation:

Indicates the value of the Count property just prior to the most recent write to the Adjust_Value property. If no such write has yet occurred, this property shall have the value zero.

Property *countBeforeChange* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.countBeforeChange

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 178

Tag Value

A 32-bit unsigned integer.

pulseConverter.countChangeTime

According to the BACnet protocol documentation:

Represents the date and time of the most recent occurrence of a write to the Adjust_Value property.

Property *countChangeTime* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.countChangeTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 179

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

pulseConverter.covIncrement

According to the BACnet protocol documentation:

Specifies the minimum change in presentValue that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

Property *covIncrement* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.covIncrement

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 22

Tag Value

A 32-bit floating point number.

pulseConverter.covPeriod

According to the BACnet protocol documentation:

Indicates the amount of time in seconds between the periodic COV notifications performed by this object.

Property *covPeriod* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.covPeriod

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 180

Tag Value

A 32-bit unsigned integer.

pulseConverter.deadband

According to the BACnet protocol documentation:

Specifies a range between the highLimit and lowLimit properties, which the presentValue must remain within for a TO-NORMAL event to be generated.

Property *deadband* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.deadband

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 25

Tag Value

A 32-bit floating point number.

pulseConverter.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

pulseConverter.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

pulseConverter.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

pulseConverter.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system

configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

pulseConverter.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *pulseConverter* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 24
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

pulseConverter.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *pulseConverter* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 24
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

pulseConverter.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *pulseConverter* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 24
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

pulseConverter.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

pulseConverter.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *pulseConverter* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 24
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

pulseConverter.highLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must exceed before an event is generated.

Property *highLimit* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.highLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 45

Tag Value

A 32-bit floating point number.

pulseConverter.inputReference

According to the BACnet protocol documentation:

Indicates the object and property (typically an Accumulator object's Present_Value property) representing the actual physical input that is to be measured and presented by the Pulse Converter object. The referenced property should have a datatype of INTEGER or Unsigned.

Property *inputReference* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.inputReference

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 181

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

pulseConverter.limitEnable

According to the BACnet protocol documentation:

Conveys two flags that separately enable and disable reporting of highLimit and lowLimit offnormal events and their return to normal.

Property *limitEnable* of object *pulseConverter* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.limitEnable
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 24
B3:	<objectAddress>
B4:	<property> = 52
Size:	2 elements

Block Elements

Index	Name	Type
0	lowLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	highLimitEnable	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

pulseConverter.lowLimit

According to the BACnet protocol documentation:

Specifies a limit that the presentValue must fall below before an event is generated.

Property *lowLimit* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.lowLimit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 59

Tag Value

A 32-bit floating point number.

pulseConverter.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

pulseConverter.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

pulseConverter.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

pulseConverter.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

pulseConverter.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

pulseConverter.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

pulseConverter.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit floating point number.

pulseConverter.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

pulseConverter.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

pulseConverter.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

pulseConverter.reliability

According to the BACnet protocol documentation:

Indicates whether the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

pulseConverter.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

pulseConverter.scaleFactor

According to the BACnet protocol documentation:

Provides the conversion factor for computing Present_Value. It represents the change in Present_Value resulting from changing the value of Count by one.

Property *scaleFactor* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.scaleFactor

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 188

Tag Value

A 32-bit floating point number.

pulseConverter.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *pulseConverter* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 24
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

pulseConverter.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

pulseConverter.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

pulseConverter.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

pulseConverter.units

According to the BACnet protocol documentation:

Indicates the measurement units of this object.

Property *units* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.units

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 117

Tag Value

An integer representing one of the options of enumeration **<BACnetEngineeringUnits>**.

pulseConverter.updateTime

According to the BACnet protocol documentation:

Indicates the date and time of the last transition of the timer state. If a transition of the timer state has never occurred, then this property shall take on the unspecified datetime value.

Property *updateTime* of object *pulseConverter* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	pulseConverter-<objectAddress>.updateTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 24
N3:	<objectAddress>
N4:	<property> = 189

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

schedule Object

According to the BACnet protocol documentation:

Defines a standardized object used to describe a periodic schedule that may recur during a range of dates, with optional exceptions at arbitrary times on arbitrary dates. The Schedule object also serves as a binding between these scheduled times and the writing of specified "values" to specific properties of specific objects at those times. Schedules are divided into days, of which there are two types: normal days within a week and exception days. Both types of days can specify scheduling events for either the full day or portions of a day, and a priority mechanism defines which scheduled event is in control at any given time.

Objects of type schedule have the following properties:

- **ackedTransitions**
- **description**
- **effectivePeriod**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **exceptionSchedule**
- **listOfObjectPropertyReferences**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityForWriting**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **scheduleDefault**
- **statusFlags**
- **tags**
- **weeklySchedule**

schedule.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *schedule* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 17
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

schedule.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

schedule.effectivePeriod

According to the BACnet protocol documentation:

Specifies the range of dates within which the Schedule object is active.

Property *effectivePeriod* of object *schedule* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.effectivePeriod
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 17
B3:	<objectAddress>
B4:	<property> = 32
Size:	2 elements

Block Elements

Index	Name	Type
0	StartDate	A Date/Time value (only the Date part).
1	EndDate	A Date/Time value (only the Date part).

schedule.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

schedule.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *schedule* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 17
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

schedule.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property `eventMessageTexts` of object `schedule` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 17
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

schedule.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property `eventMessageTextsConfig` of object `schedule` can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 17
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

schedule.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

schedule.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *schedule* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 17
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

schedule.exceptionSchedule

According to the BACnet protocol documentation:

An array describing a sequence of scheduled actions that takes precedence over the normal day's behavior on a specific day or days.

Property *exceptionSchedule* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.exceptionSchedule

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 38

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetSpecialEvent>**.

schedule.listOfObjectPropertyReferences

According to the BACnet protocol documentation:

Specifies the Object and Property identifiers of the properties to be written with specific values at specific times on specific days.

Property *listOfObjectPropertyReferences* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.listOfObjectPropertyReferences

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 54

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectPropertyReference>**.

schedule.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

schedule.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

schedule.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

schedule.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

schedule.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

schedule.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

schedule.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A simple value type (Integer, Real, Date, String) or a string containing an XML representation of: Null, Boolean, Unsigned, INTEGER, REAL, Double, OctetString, CharacterString, BitString, Enumerated, Date, Time, BACnetObjectIdentifier.

schedule.priorityForWriting

According to the BACnet protocol documentation:

Provides a priority to be used by the command prioritization mechanism.

Property *priorityForWriting* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.priorityForWriting

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 88

Tag Value

A 32-bit unsigned integer.

schedule.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

schedule.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

schedule.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

schedule.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

schedule.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

schedule.scheduleDefault

According to the BACnet protocol documentation:

This property holds a default value to be used for the presentValue property when no other scheduled value is in effect.

Property *scheduleDefault* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.scheduleDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 174

Tag Value

A simple value type (Integer, Real, Date, String) or a string containing an XML representation of: Null, Boolean, Unsigned, INTEGER, REAL, Double, OctetString, CharacterString, BitString, Enumerated, Date, Time, BACnetObjectIdentifier.

schedule.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *schedule* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 17
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

schedule.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *schedule* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 17
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

schedule.weeklySchedule

According to the BACnet protocol documentation:

An array containing the schedule of a week day (each pair of elements contains a time and a setpoint value).

Property *weeklySchedule* of object *schedule* can be read and written using a block tag with 7 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	schedule-<objectAddress>.weeklySchedule
Size	7 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 17
B3:	<objectAddress>
B4:	<property> = 123
Size:	7 elements

Block Elements

Index	Name	Type
0	elm00	A character string containing an XML with a value of type <BACnetDailySchedule> .
1	elm01	A character string containing an XML with a value of type <BACnetDailySchedule> .
2	elm02	A character string containing an XML with a value of type <BACnetDailySchedule> .
3	elm03	A character string containing an XML with a value of type <BACnetDailySchedule> .
4	elm04	A character string containing an XML with a value of type <BACnetDailySchedule> .
5	elm05	A character string containing an XML with a value of type <BACnetDailySchedule> .
6	elm06	A character string containing an XML with a value of type <BACnetDailySchedule> .

structuredView Object

According to the BACnet protocol documentation:

Defines a standardized object that provides a container to hold references to subordinate objects, which may include other Structured View objects, thereby allowing multilevel hierarchies to be created. The hierarchies are intended to convey a structure or organization such as a geographical distribution or application organization. Subordinate objects may reside in the same device as the Structured View object or in other devices on the network.

Objects of type structuredView have the following properties:

- **defaultSubordinateRelationship**
- **description**
- **nodeSubtype**
- **nodeType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **profileLocation**
- **profileName**
- **propertyList**
- **represents**
- **subordinateAnnotations**
- **subordinateList**
- **subordinateNodeTypes**
- **subordinateRelationships**
- **subordinateTags**
- **tags**

structuredView.defaultSubordinateRelationship

According to the BACnet protocol documentation:

Describes the default relationship to each member of the Subordinate_List, unless overridden by a member of the Subordinate_Relationships property. If this property is absent and the Subordinate_Relationships is also absent, the relationship to the subordinates is equal to UNKNOWN.

Property *defaultSubordinateRelationship* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.defaultSubordinateRelationship

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 490

Tag Value

An integer representing one of the options of enumeration **<BACnetRelationship>**.

structuredView.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

structuredView.nodeSubtype

According to the BACnet protocol documentation:

Contains a string of printable characters whose content is not restricted. It provides a more specific classification of the object in the hierarchy of objects, providing a short description of the item represented by the node.

Property *nodeSubtype* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.nodeSubtype

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 207

Tag Value

A string value.

structuredView.nodeType

According to the BACnet protocol documentation:

Provides a general classification of the object in the hierarchy of objects. It is intended as a general suggestion to a client application about the contents of a Structured View object, and is not intended to convey an exact definition. Further refinement of classification is provided by the Node_Subtype property.

Property *nodeType* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.nodeType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 208

Tag Value

An integer representing one of the options of enumeration **<BACnetNodeType>**.

structuredView.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

structuredView.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

structuredView.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

structuredView.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

structuredView.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

structuredView.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

structuredView.represents

According to the BACnet protocol documentation:

Used to indicate the entity for which this Structured View is modeling the subordinates.

Property *represents* of object *structuredView* can be read and written using a block tag with 2 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.represents
Size	2 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 29
B3:	<objectAddress>
B4:	<property> = 491
Size:	2 elements

Block Elements

Index	Name	Type
0	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.
1	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

structuredView.subordinateAnnotations

According to the BACnet protocol documentation:

Used to define a text string description for each member of the Subordinate_List. The content of these strings is not restricted. If the size of this array is changed, the size of the Subordinate_List, Subordinate_Tags, Subordinate_Relationships, and Subordinate_Node_Types arrays shall also be changed to the same size.

Property *subordinateAnnotations* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.subordinateAnnotations

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 210

Tag Value

A character string containing an XML with a value of type **<SequenceOfCharacterString>**.

structuredView.subordinateList

According to the BACnet protocol documentation:

Defines the members of the current Structured View.

Property *subordinateList* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.subordinateList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 211

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectReference>**.

structuredView.subordinateNodeTypes

According to the BACnet protocol documentation:

Provides node type information for each member of the Subordinate_List. If the subordinate object has its own Node_Type, it is intended that the local value in this property logically override the subordinate's Node_Type unless the local value is "UNKNOWN", in which case, the subordinate's Node_Type is used. If the size of this array is changed, the size of the Subordinate_List, Subordinate_Annotations, Subordinate_Tags, and Subordinate_Relationships arrays, if present, shall also be changed to the same size. Uninitialized Subordinate_Node_Types array elements shall have the value UNKNOWN.

Property *subordinateNodeTypes* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.subordinateNodeTypes

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 487

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNodeType>**.

structuredView.subordinateRelationships

According to the BACnet protocol documentation:

Describes the relationship to each member of the Subordinate_List. If this property is absent, then relationship to each of the subordinates is equal to the value of the Default_Subordinate_Relationship property, if present, else equal to UNKNOWN. If the size of this array is changed, the size of the Subordinate_List, Subordinate_Annotations, Subordinate_Tags, and Subordinate_Node_Types arrays, if present, shall also be changed to the same size. Uninitialized Subordinate_Relationships array elements shall be equal to DEFAULT.

Property *subordinateRelationships* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.subordinateRelationships

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 489

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetRelationship>**.

structuredView.subordinateTags

According to the BACnet protocol documentation:

Provides individual collections of tags for each of the subordinates. If the size of this array is changed, the size of the Subordinate_List, Subordinate_Annotations, Subordinate_Node_Types, and Subordinate_Relationships arrays, if present, shall also be changed to the same size. Uninitialized Subordinate_Tags array elements shall be empty collections.

Property *subordinateTags* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.subordinateTags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 488

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValueCollection>**.

structuredView.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *structuredView* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	structuredView-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 29
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

timeValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a Time Value object to make any kind of time data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client. A Time Value object is used to represent a single moment in time. In contrast, the Time Pattern Value object can be used to represent multiple recurring times. Time Value objects may optionally support intrinsic reporting to facilitate the reporting of fault conditions.

Objects of type timeValue have the following properties:

- **ackedTransitions**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **lastCommandTime**
- **notificationClass**
- **notifyType**
- **objectIdentifier**

- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **statusFlags**
- **tags**
- **valueSource**
- **valueSourceArray**

timeValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *timeValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 50
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timeValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *timeValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 50
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

		• A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
7	elm07	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
8	elm08	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

		• A character string containing an XML with a value of type <BACnetTimeStamp>.
9	elm09	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
10	elm10	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
11	elm11	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
12	elm12	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
13	elm13	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
14	elm14	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
15	elm15	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set;

- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

timeValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

timeValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

timeValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timeValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *timeValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 50
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timeValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *timeValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 50
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

timeValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *timeValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 50
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

timeValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

timeValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *timeValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 50
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

timeValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

timeValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

timeValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

timeValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

timeValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

timeValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

timeValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timeValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A Date/Time value (only the Time part).

timeValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *timeValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 50
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
1	elm01	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
2	elm02	One of the following data types:

timeValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

timeValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

timeValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

timeValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

timeValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timeValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A Date/Time value (only the Time part).

timeValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *timeValue* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 50
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timeValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

timeValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *timeValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 50
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

timeValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *timeValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timeValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 50
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

timepatternValue Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a Time Pattern Value object to make any kind of time data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client. Time Pattern objects can be used to represent multiple recurring times based on rules defined by the pattern of individual fields of the time, some of which may be "don't care", which matches any value in that field. Examples of possibilities would be: "every minute of the 11 o'clock hour of the day", or "the thirteenth minute of any hour". Time Pattern Value objects may optionally support intrinsic reporting to facilitate the reporting of fault conditions.

Objects of type `timepatternValue` have the following properties:

- **ackedTransitions**
- **commandTimeArray**
- **currentCommandPriority**
- **description**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **lastCommandTime**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityArray**
- **profileLocation**
- **profileName**
- **propertyList**
- **reliability**
- **reliabilityEvaluationInhibit**
- **relinquishDefault**
- **statusFlags**
- **tags**

- **valueSource**
- **valueSourceArray**

timepatternValue.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *timepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 49
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timepatternValue.commandTimeArray

According to the BACnet protocol documentation:

Indicates the time at which each priority was last commanded or relinquished.

Property *commandTimeArray* of object *timepatternValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.commandTimeArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 49
B3:	<objectAddress>
B4:	<property> = 430
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
3	elm03	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
4	elm04	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
5	elm05	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
6	elm06	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number;

timepatternValue.currentCommandPriority

According to the BACnet protocol documentation:

Indicates the currently active priority. The value of this property shall be equal to the index of the entry in the Priority_Array from which the Present_Value's value has been taken. If Present_Value has taken on the value of Relinquish_Default, this property shall have the value Null.

Property *currentCommandPriority* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.currentCommandPriority

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 431

Tag Value

One of the following data types:

- A null/empty value.
- A 32-bit unsigned integer.

timepatternValue.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

timepatternValue.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timepatternValue.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *timepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 49
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timepatternValue.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *timepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 49
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

timepatternValue.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *timepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 49
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

timepatternValue.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

timepatternValue.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *timepatternValue* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 49
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

timepatternValue.lastCommandTime

According to the BACnet protocol documentation:

Indicates the time at which the Present_Value, Current_Command_Priority, or Value_Source last changed.

Property *lastCommandTime* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.lastCommandTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 432

Tag Value

A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet.

- A null value means the timestamp is empty/not set;
- A date/time value representing a precise date and time;
- An integer defining the timestamp sequence number;
- A character string containing an XML with a value of type **<BACnetTimeStamp>**.

timepatternValue.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

timepatternValue.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

timepatternValue.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

timepatternValue.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

timepatternValue.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

timepatternValue.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timepatternValue.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A Date/Time value (only the Time part).

timepatternValue.priorityArray

According to the BACnet protocol documentation:

Read-only array of prioritized values.

Property *priorityArray* of object *timepatternValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.priorityArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 49
B3:	<objectAddress>
B4:	<property> = 87
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
1	elm01	One of the following data types: • A null/empty value. • A 32-bit floating point number. • An unsigned integer representing an enumerated value. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <Boolean>. • A character string containing an XML with a value of type <INTEGER>. • A character string containing an XML with a value of type <Double>. • A Date/Time value (only the Time part). • A string value. • A character string containing an XML with a value of type <OctetString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Date>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>.
2	elm02	One of the following data types:

timepatternValue.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

timepatternValue.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

timepatternValue.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

timepatternValue.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

timepatternValue.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timepatternValue.relinquishDefault

According to the BACnet protocol documentation:

The default value to be used for the presentValue property when all command priority values in the priorityArray property have a NULL value.

Property *relinquishDefault* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.relinquishDefault

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 104

Tag Value

A Date/Time value (only the Time part).

timepatternValue.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *timepatternValue* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 49
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timepatternValue.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

timepatternValue.valueSource

According to the BACnet protocol documentation:

Indicates the source of the value of the Mode property.

Property *valueSource* of object *timepatternValue* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.valueSource

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 49
N3:	<objectAddress>
N4:	<property> = 433

Tag Value

One of the following data types:

- A null/empty value.
- A character string containing an XML with a value of type **<BACnetDeviceObjectReference>**.
- A character string containing an XML with a value of type **<BACnetAddress>**.

timepatternValue.valueSourceArray

According to the BACnet protocol documentation:

Indicates the source of the last command at each priority.

Property *valueSourceArray* of object *timepatternValue* can be read and written using a block tag with 16 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timepatternValue-<objectAddress>.valueSourceArray
Size	16 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 49
B3:	<objectAddress>
B4:	<property> = 434
Size:	16 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
1	elm01	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
2	elm02	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
3	elm03	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
4	elm04	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
5	elm05	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
6	elm06	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
7	elm07	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
8	elm08	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
9	elm09	One of the following data types: • A null/empty value.

		• A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
10	elm10	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
11	elm11	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
12	elm12	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
13	elm13	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
14	elm14	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.
15	elm15	One of the following data types: • A null/empty value. • A character string containing an XML with a value of type <BACnetDeviceObjectReference>. • A character string containing an XML with a value of type <BACnetAddress>.

timer Object

According to the BACnet protocol documentation:

Defines a standardized object whose properties represent the externally visible characteristics of a countdown timer. The Timer object provides a network-visible view of selected parameters of a countdown timer. The operating state of the timer may be viewed and controlled through these properties.

Objects of type timer have the following properties:

- **ackedTransitions**
- **alarmValues**
- **defaultTimeout**
- **description**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **expirationTime**
- **initialTimeout**
- **lastStateChange**
- **listOfObjectPropertyReferences**
- **maxPresValue**
- **minPresValue**
- **notificationClass**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **outOfService**
- **presentValue**
- **priorityForWriting**
- **profileLocation**
- **profileName**
- **propertyList**

- **reliability**
- **reliabilityEvaluationInhibit**
- **resolution**
- **stateChangeValues**
- **statusFlags**
- **tags**
- **timeDelay**
- **timeDelayNormal**
- **timerRunning**
- **timerState**
- **updateTime**

timer.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *timer* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 31
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timer.alarmValues

According to the BACnet protocol documentation:

Specifies any states the presentValue must equal before a TO-OFFNORMAL event is generated.

Property *alarmValues* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.alarmValues

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 7

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetTimerState>**.

timer.defaultTimeout

According to the BACnet protocol documentation:

Specifies the default initial timeout, in milliseconds. This default timeout is used as the initial timeout when a value of TRUE is written to the Timer_Running property. The value of this property shall be within the supported range specified by the values of the Min_Pres_Value and Max_Pres_Value properties.

Property *defaultTimeout* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.defaultTimeout

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 393

Tag Value

A 32-bit unsigned integer.

timer.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

timer.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timer.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

timer.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timer.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *timer* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 31
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timer.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *timer* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 31
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

timer.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *timer* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 31
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

timer.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

timer.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *timer* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 31
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

timer.expirationTime

According to the BACnet protocol documentation:

Specifies the date and time after which the credential will expire. This defines the end of the validity period of the credential. If the current time is after the expiry time, the credential shall be disabled and the value `DISABLED_EXPIRED` shall be added to the `Reason_For_Disable` list. The value `DISABLED_EXPIRED` shall be removed from the list when this condition no longer applies. If all of the fields of the `BACnetDateTime` value contain a value of `X'FF'`, then the credential has an expiry time of 'end-of-time'.

Property `expirationTime` of object `timer` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.expirationTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 270

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

timer.initialTimeout

According to the BACnet protocol documentation:

Indicates the initial timeout, in milliseconds, that was taken as initial duration to count down when the timer last transitioned to state `RUNNING`.

Property `initialTimeout` of object `timer` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.initialTimeout

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 394

Tag Value

A 32-bit unsigned integer.

timer.lastStateChange

According to the BACnet protocol documentation:

Indicates the last transition the timer state machine performed.

Property *lastStateChange* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.lastStateChange

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 395

Tag Value

An integer representing one of the options of enumeration **<BACnetTimerTransition>**.

timer.listOfObjectPropertyReferences

According to the BACnet protocol documentation:

Specifies the Object and Property identifiers of the properties to be written with specific values at specific times on specific days.

Property *listOfObjectPropertyReferences* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.listOfObjectPropertyReferences

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 54

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectPropertyReference>**.

timer.maxPresValue

According to the BACnet protocol documentation:

Indicates the highest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *maxPresValue* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.maxPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 65

Tag Value

A 32-bit unsigned integer.

timer.minPresValue

According to the BACnet protocol documentation:

Indicates the lowest number in engineering units that can be reliably obtained for the presentValue property of this object.

Property *minPresValue* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.minPresValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 69

Tag Value

A 32-bit unsigned integer.

timer.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

timer.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

timer.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

timer.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

timer.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

timer.outOfService

According to the BACnet protocol documentation:

Indicates weather (TRUE) or not (FALSE) the physical input that the object represents is not in service.

Property *outOfService* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.outOfService

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 81

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timer.presentValue

According to the BACnet protocol documentation:

Indicates the current value of the object.

Property *presentValue* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.presentValue

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 85

Tag Value

A 32-bit unsigned integer.

timer.priorityForWriting

According to the BACnet protocol documentation:

Provides a priority to be used by the command prioritization mechanism.

Property *priorityForWriting* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.priorityForWriting

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 88

Tag Value

A 32-bit unsigned integer.

timer.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

timer.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

timer.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

timer.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

timer.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timer.resolution

According to the BACnet protocol documentation:

Indicates the smallest recognizable change in presentValue in engineering units (read-only).

Property *resolution* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.resolution

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 106

Tag Value

A 32-bit unsigned integer.

timer.stateChangeValues

According to the BACnet protocol documentation:

Represents the values that are to be written to the referenced properties when a change of the timer state occurs. Each of the elements 1-7 of this array may contain a value to be written for the respective change of timer state. The array index of the element is equal to the numerical value of the BACnetTimerTransition enumeration for the respective timer state change. The timer state change NONE has no corresponding array element.

Property *stateChangeValues* of object *timer* can be read and written using a block tag with 7 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.stateChangeValues
Size	7 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 31
B3:	<objectAddress>
B4:	<property> = 396
Size:	7 elements

Block Elements

Index	Name	Type
0	elm00	One of the following data types: • A null/empty value. • An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <INTEGER>. • A 32-bit floating point number. • A character string containing an XML with a value of type <Double>. • A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A'). • A character string containing an XML with a value of type <CharacterString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Enumerated>. • A Date/Time value (only the Date part). • A character string containing an XML with a value of type <Time>. • A character string containing an XML with a value of type <BACnetObjectIdentifier>. • A character string containing an XML with a value of type <Null>. • A character string containing an XML with a value of type <Any>. • A character string containing an XML with a value of type <BACnetDateTime>. • A character string containing an XML with a value of type <BACnetLightingCommand>.
1	elm01	One of the following data types: • A null/empty value. • An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE. • A character string containing an XML with a value of type <Unsigned>. • A character string containing an XML with a value of type <INTEGER>. • A 32-bit floating point number. • A character string containing an XML with a value of type <Double>. • A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A'). • A character string containing an XML with a value of type <CharacterString>. • A character string containing an XML with a value of type <BitString>. • A character string containing an XML with a value of type <Enumerated>. • A Date/Time value (only the Date part).

- A character string containing an XML with a value of type **<Time>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Null>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.
- A character string containing an XML with a value of type **<BACnetLightingCommand>**.

2

elm02

One of the following data types:

- A null/empty value.
- An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A 32-bit floating point number.
- A character string containing an XML with a value of type **<Double>**.
- A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').
- A character string containing an XML with a value of type **<CharacterString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Enumerated>**.
- A Date/Time value (only the Date part).
- A character string containing an XML with a value of type **<Time>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Null>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.
- A character string containing an XML with a value of type **<BACnetLightingCommand>**.

3

elm03

One of the following data types:

- A null/empty value.
- An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A 32-bit floating point number.
- A character string containing an XML with a value of type **<Double>**.

- A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').
- A character string containing an XML with a value of type **<CharacterString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Enumerated>**.
- A Date/Time value (only the Date part).
- A character string containing an XML with a value of type **<Time>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Null>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.
- A character string containing an XML with a value of type **<BACnetLightingCommand>**.

4

elm04

One of the following data types:

- A null/empty value.
- An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A 32-bit floating point number.
- A character string containing an XML with a value of type **<Double>**.
- A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').
- A character string containing an XML with a value of type **<CharacterString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Enumerated>**.
- A Date/Time value (only the Date part).
- A character string containing an XML with a value of type **<Time>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Null>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.
- A character string containing an XML with a value of type **<BACnetLightingCommand>**.

5

elm05

One of the following data types:

- A null/empty value.

- An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A 32-bit floating point number.
- A character string containing an XML with a value of type **<Double>**.
- A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').
- A character string containing an XML with a value of type **<CharacterString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Enumerated>**.
- A Date/Time value (only the Date part).
- A character string containing an XML with a value of type **<Time>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Null>**.
- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.
- A character string containing an XML with a value of type **<BACnetLightingCommand>**.

6

elm06

One of the following data types:

- A null/empty value.
- An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
- A character string containing an XML with a value of type **<Unsigned>**.
- A character string containing an XML with a value of type **<INTEGER>**.
- A 32-bit floating point number.
- A character string containing an XML with a value of type **<Double>**.
- A string value representing a sequence of octets, where each octet is represented by a pair of hexadecimal characters (ex: '0F121A').
- A character string containing an XML with a value of type **<CharacterString>**.
- A character string containing an XML with a value of type **<BitString>**.
- A character string containing an XML with a value of type **<Enumerated>**.
- A Date/Time value (only the Date part).
- A character string containing an XML with a value of type **<Time>**.
- A character string containing an XML with a value of type **<BACnetObjectIdentifier>**.
- A character string containing an XML with a value of type **<Null>**.

- A character string containing an XML with a value of type **<Any>**.
- A character string containing an XML with a value of type **<BACnetDateTime>**.
- A character string containing an XML with a value of type **<BACnetLightingCommand>**.

timer.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *timer* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 31
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timer.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

timer.timeDelay

According to the BACnet protocol documentation:

Specifies the minimum period of time in seconds that the presentValue must remain outside the band defined by the highLimit and lowLimit properties before a TO_OFFNORMAL event is generated or within the same band, including the deadband property, before a TO_NORMAL event is generated.

Property *timeDelay* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.timeDelay

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 113

Tag Value

A 32-bit unsigned integer.

timer.timeDelayNormal

According to the BACnet protocol documentation:

Contains the pTimeDelayNormal parameter for the object's event algorithm.

Property *timeDelayNormal* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.timeDelayNormal

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 356

Tag Value

A 32-bit unsigned integer.

timer.timerRunning

According to the BACnet protocol documentation:

Contains a value of TRUE if the current state of the timer is RUNNING, otherwise FALSE. This property may be used by other objects that require a simple BOOLEAN flag for determining if the timer is in RUNNING state. Writing a value of TRUE to this property, in any timer state, shall be considered a start request. Present_Value shall be set to the value specified in the Default_Timeout property. Writing a value of FALSE to this property while the timer is in the RUNNING state shall be considered an expire request and shall force the timer to transition to state EXPIRED. When writing a value of FALSE to this property while the timer is in the EXPIRED or IDLE state, no transition of the timer state shall occur.

Property *timerRunning* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.timerRunning

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 397

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

timer.timerState

According to the BACnet protocol documentation:

Indicates the current state of the timer this object represents. To clear the timer, i.e. to request the timer to enter the IDLE state, a value of IDLE is written to this property.

Property *timerState* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.timerState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 398

Tag Value

An integer representing one of the options of enumeration **<BACnetTimerState>**.

timer.updateTime

According to the BACnet protocol documentation:

Indicates the date and time of the last transition of the timer state. If a transition of the timer state has never occurred, then this property shall take on the unspecified datetime value.

Property *updateTime* of object *timer* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	timer-<objectAddress>.updateTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 31
N3:	<objectAddress>
N4:	<property> = 189

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

trendLog Object

According to the BACnet protocol documentation:

Monitors a property of a referenced object and, when predefined conditions are met, saves ("logs") the value of the property and a timestamp in an internal log buffer for subsequent retrieval. The data may be logged periodically, upon a change of value or when "triggered" by a write to the Trigger property. The Trigger property allows the acquisition of samples to be controlled by network write operations or internal processes. Errors that prevent the acquisition of the data, as well as changes in the status or operation of the logging process itself, are also recorded. Each timestamped buffer entry is called a "log record."

Objects of type trendLog have the following properties:

- **ackedTransitions**
- **alignIntervals**
- **bufferSize**
- **clientCovIncrement**
- **covResubscriptionInterval**
- **description**
- **enable**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **intervalOffset**
- **lastNotifyRecord**
- **logBuffer**
- **logDeviceObjectProperty**
- **logInterval**
- **loggingType**

- **notificationClass**
- **notificationThreshold**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **profileLocation**
- **profileName**
- **propertyList**
- **recordCount**
- **recordsSinceNotification**
- **reliability**
- **reliabilityEvaluationInhibit**
- **startTime**
- **statusFlags**
- **stopTime**
- **stopWhenFull**
- **tags**
- **totalRecordCount**
- **trigger**

trendLog.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *trendLog* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 20
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLog.alignIntervals

According to the BACnet protocol documentation:

Specifies whether (TRUE) or not (FALSE) clock-aligned periodic time synchronizations is enabled.

Property *alignIntervals* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.alignIntervals

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 193

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLog.bufferSize

According to the BACnet protocol documentation:

Specifies the maximum number of log records the log buffer may hold. If writable, it may not be written when Enable is TRUE. The disposition of existing log records when Buffer_Size is written is a local matter. If all records are deleted when the Buffer_Size is written then the object shall act as if the Record_Count was set to zero.

Property *bufferSize* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.bufferSize

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 126

Tag Value

A 32-bit unsigned integer.

trendLog.clientCovIncrement

According to the BACnet protocol documentation:

Specifies the increment to be used in determining that a change of value has occurred.

Property *clientCovIncrement* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.clientCovIncrement

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 127

Tag Value

One of the following data types:

- A 32-bit floating point number.
- A null/empty value.

trendLog.covResubscriptionInterval

According to the BACnet protocol documentation:

Specifies the number of seconds between COV resubscriptions, provided that COV subscription is in effect.

Property *covResubscriptionInterval* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.covResubscriptionInterval

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 128

Tag Value

A 32-bit unsigned integer.

trendLog.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

trendLog.enable

According to the BACnet protocol documentation:

Indicates and controls whether (TRUE) or not (FALSE) logging of events is enabled. Logging occurs if and only if Enable is TRUE, Local_Time is on or after Start_Time, and Local_Time is before Stop_Time. If Start_Time contains an unspecified datetime, then it shall be considered equal to 'the start of time'. If Stop_Time contains an unspecified datetime, then it shall be considered equal to 'the end of time'. Log records of type log-status are recorded without regard to the value of the Enable property.

Property *enable* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.enable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 133

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLog.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLog.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

trendLog.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLog.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *trendLog* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 20
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLog.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *trendLog* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 20
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

trendLog.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *trendLog* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 20
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

trendLog.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

trendLog.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *trendLog* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 20
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

trendLog.intervalOffset

According to the BACnet protocol documentation:

Specifies the offset in minutes from the beginning of the period specified for time synchronization until the actual time synchronization requests are sent.

Property *intervalOffset* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.intervalOffset

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 195

Tag Value

A 32-bit unsigned integer.

trendLog.lastNotifyRecord

According to the BACnet protocol documentation:

Represents the pPreviousCount parameter of the object's event algorithm.

Property *lastNotifyRecord* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.lastNotifyRecord

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 173

Tag Value

A 32-bit unsigned integer.

trendLog.logBuffer

According to the BACnet protocol documentation:

Contains a list of up to Buffer_Size timestamped log records of datatype BACnetEventLogRecord, each of which conveys the event notification parameters or status changes in the Event Log object.

Property *logBuffer* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.logBuffer

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 131

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLogRecord>**.

trendLog.logDeviceObjectProperty

According to the BACnet protocol documentation:

Specifies the properties to be logged.

Property *logDeviceObjectProperty* of object *trendLog* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.logDeviceObjectProperty
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 20
B3:	<objectAddress>
B4:	<property> = 132
Size:	4 elements

Block Elements

Index	Name	Type
0	ObjectIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').
1	PropertyIdentifier	An integer representing one of the options of enumeration <BACnetPropertyIdentifier> .
2	PropertyArrayIndex	A 32-bit unsigned integer. OBS: this is an optional element, a <null> value indicates the element is not present.
3	DeviceIdentifier	A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345'). OBS: this is an optional element, a <null> value indicates the element is not present.

trendLog.logInterval

According to the BACnet protocol documentation:

Specifies the periodic interval in hundredths of seconds for which the referenced properties are to be logged when Logging_Type has the value POLLED. If Logging_Type has the value TRIGGERED, then the value of this property shall be zero and ignored. This property shall be writable if Logging_Type has the value POLLED, and shall be read-only if Logging_Type has the value TRIGGERED.

Property *logInterval* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.logInterval

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 134

Tag Value

A 32-bit unsigned integer.

trendLog.loggingType

According to the BACnet protocol documentation:

Specifies whether the Trend Log Multiple object collects log records using polling or triggered acquisition. COV Logging is not allowed for a Trend Log Multiple object.

Property *loggingType* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.loggingType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 197

Tag Value

An integer representing one of the options of enumeration **<BACnetLoggingType>**.

trendLog.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

trendLog.notificationThreshold

According to the BACnet protocol documentation:

*Contains the *pThreshold* parameter for the object's event algorithm.*

Property *notificationThreshold* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.notificationThreshold

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 137

Tag Value

A 32-bit unsigned integer.

trendLog.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

trendLog.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

trendLog.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

trendLog.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

trendLog.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

trendLog.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

trendLog.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

trendLog.recordCount

According to the BACnet protocol documentation:

Represents the number of records currently resident in the log buffer.

Property *recordCount* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.recordCount

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 141

Tag Value

A 32-bit unsigned integer.

trendLog.recordsSinceNotification

According to the BACnet protocol documentation:

Represents the number of log records collected since the previous notification, or since the beginning of logging if no previous notification has occurred.

Property *recordsSinceNotification* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.recordsSinceNotification

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 140

Tag Value

A 32-bit unsigned integer.

trendLog.reliability

According to the BACnet protocol documentation:

Indicates whether the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

trendLog.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLog.startTime

According to the BACnet protocol documentation:

Specifies the date and time at or after which logging shall be enabled by this property. If this property contains an unspecified datetime, then the conditions for logging to be enabled by Start_Time shall be ignored. If Start_Time specifies a date and time after Stop_Time, then logging shall be disabled. This property shall be writable if present. When Start_Time is reached, the value of the Enable property is not changed.

Property *startTime* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.startTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 142

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

trendLog.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *trendLog* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 20
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLog.stopTime

According to the BACnet protocol documentation:

Specifies the date and time at or after which logging shall be disabled by this property. If this property contains an unspecified datetime, then the conditions for logging to be disabled by Stop_Time shall be ignored. If Stop_Time specifies a date and time earlier than Start_Time, then logging shall be disabled. This property shall be writable if present. When Stop_Time is reached, the value of the Enable property is not changed.

Property *stopTime* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.stopTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 143

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;

- A character string containing an XML with a value of type **<BACnetDateTime>**.

trendLog.stopWhenFull

According to the BACnet protocol documentation:

Specifies whether (TRUE) or not (FALSE) logging should cease when the log buffer is full. When logging ceases because the addition of one more log record would cause the log buffer to be full, Enable shall be set to FALSE and the event recorded. If Stop_When_Full is writable, attempts to write the value TRUE to the Stop_When_Full property while Record_Count is equal to Buffer_Size shall result in the oldest log record in the log buffer being discarded, and shall cause the Enable property to be set to FALSE and the event to be recorded.

Property *stopWhenFull* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.stopWhenFull

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 144

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLog.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property *tags* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

trendLog.totalRecordCount

According to the BACnet protocol documentation:

Represents the total number of log records collected by the Trend Log object since creation. When the value of Total_Record_Count reaches its maximum possible value of 232 - 1, the next value it takes shall be one. Once this value has wrapped to one, its semantic value (the total number of log records collected) has been lost but its use in generating notifications remains.

Property *totalRecordCount* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.totalRecordCount

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 145

Tag Value

A 32-bit unsigned integer.

trendLog.trigger

According to the BACnet protocol documentation:

Causes the Trend Log object to acquire a log record whenever the value of this property is changed from FALSE to TRUE. It shall remain TRUE while the Trend Log object is acquiring the data items for a log record. When all data items have been collected or it has been determined that all outstanding data requests will not be fulfilled, the Trend Log object shall reset the value to FALSE.

Property *trigger* of object *trendLog* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLog-<objectAddress>.trigger

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 20
N3:	<objectAddress>
N4:	<property> = 205

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLogMultiple Object

According to the BACnet protocol documentation:

Monitors one or more properties of one or more referenced objects, either in the same device as the Trend Log Multiple object or in an external device. When predefined conditions are met, the object saves ("logs") the value of the properties and a timestamp into an internal log buffer for subsequent retrieval. The data may be logged periodically or when "triggered" by a write to the Trigger property. Errors that prevent the acquisition of the data, as well as changes in the status or operation of the logging process itself, are also recorded. Each timestamped log buffer entry is called a "log record".

Objects of type trendLogMultiple have the following properties:

- **ackedTransitions**
- **alignIntervals**
- **bufferSize**
- **description**
- **enable**
- **eventAlgorithmInhibit**
- **eventAlgorithmInhibitRef**
- **eventDetectionEnable**
- **eventEnable**
- **eventMessageTexts**
- **eventMessageTextsConfig**
- **eventState**
- **eventTimeStamps**
- **intervalOffset**
- **lastNotifyRecord**

- **logBuffer**
- **logDeviceObjectProperty**
- **logInterval**
- **loggingType**
- **notificationClass**
- **notificationThreshold**
- **notifyType**
- **objectIdentifier**
- **objectName**
- **objectType**
- **profileLocation**
- **profileName**
- **propertyList**
- **recordCount**
- **recordsSinceNotification**
- **reliability**
- **reliabilityEvaluationInhibit**
- **startTime**
- **statusFlags**
- **stopTime**
- **stopWhenFull**
- **tags**
- **totalRecordCount**
- **trigger**

trendLogMultiple.ackedTransitions

According to the BACnet protocol documentation:

Conveys three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

Property *ackedTransitions* of object *trendLogMultiple* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.ackedTransitions
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 27
B3:	<objectAddress>
B4:	<property> = 0
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLogMultiple.alignIntervals

According to the BACnet protocol documentation:

Specifies whether (TRUE) or not (FALSE) clock-aligned periodic time synchronizations is enabled.

Property *alignIntervals* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.alignIntervals

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 193

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLogMultiple.bufferSize

According to the BACnet protocol documentation:

Specifies the maximum number of log records the log buffer may hold. If writable, it may not be written when Enable is TRUE. The disposition of existing log records when Buffer_Size is written is a local matter. If all records are deleted when the Buffer_Size is written then the object shall act as if the Record_Count was set to zero.

Property *bufferSize* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.bufferSize

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 126

Tag Value

A 32-bit unsigned integer.

trendLogMultiple.description

According to the BACnet protocol documentation:

String describing the object.

Property *description* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.description

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 28

Tag Value

A string value.

trendLogMultiple.enable

According to the BACnet protocol documentation:

Indicates and controls whether (TRUE) or not (FALSE) logging of events is enabled. Logging occurs if and only if Enable is TRUE, Local_Time is on or after Start_Time, and Local_Time is before Stop_Time. If Start_Time contains an unspecified datetime, then it shall be considered equal to 'the start of time'. If Stop_Time contains an unspecified datetime, then it shall be considered equal to 'the end of time'. Log records of type log-status are recorded without regard to the value of the Enable property.

Property *enable* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.enable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 133

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLogMultiple.eventAlgorithmInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) the event algorithm has been disabled for the object. This property is a runtime override that allows temporary disabling of the event algorithm. If the Event_Algorithm_Inhibit_Ref property is present and initialized (contains an instance other than 4194303), then the Event_Algorithm_Inhibit property shall be read-only and shall reflect the value of the property referenced by Event_Algorithm_Inhibit_Ref. A BACnetBinaryPV value of INACTIVE shall map to a value of FALSE and a value of ACTIVE shall map to a value of TRUE. If the referenced property does not exist, it shall be assumed to have a value of FALSE.

Property *eventAlgorithmInhibit* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.eventAlgorithmInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 354

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLogMultiple.eventAlgorithmInhibitRef

According to the BACnet protocol documentation:

Indicates the property which controls the value of property Event_Algorithm_Inhibit. When this property is present and initialized (contains an instance other than 4194303), the referenced property shall be of type BACnetBinaryPV or BOOLEAN.

Property *eventAlgorithmInhibitRef* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.eventAlgorithmInhibitRef

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 355

Tag Value

A string representing an object and a property in the syntax: '<objectType>-<objectAddress>.<property>'. An optional array index '(<index>)' can be used on array properties.

trendLogMultiple.eventDetectionEnable

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services. This property is expected to be set during system configuration and is not expected to change dynamically. When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

Property *eventDetectionEnable* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.eventDetectionEnable

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 353

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLogMultiple.eventEnable

According to the BACnet protocol documentation:

Conveys three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Analog Input objects, transitions to High_Limit and Low_Limit Event_States are considered to be 'offnormal' events.

Property *eventEnable* of object *trendLogMultiple* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.eventEnable
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 27
B3:	<objectAddress>
B4:	<property> = 35
Size:	3 elements

Block Elements

Index	Name	Type
0	toOffnormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	toFault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	toNormal	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLogMultiple.eventMessageTexts

According to the BACnet protocol documentation:

Conveys the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

Property *eventMessageTexts* of object *trendLogMultiple* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.eventMessageTexts
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 27
B3:	<objectAddress>
B4:	<property> = 351
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

trendLogMultiple.eventMessageTextsConfig

According to the BACnet protocol documentation:

Contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

Property *eventMessageTextsConfig* of object *trendLogMultiple* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.eventMessageTextsConfig
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 27
B3:	<objectAddress>
B4:	<property> = 352
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A string value.
1	elm01	A string value.
2	elm02	A string value.

trendLogMultiple.eventState

According to the BACnet protocol documentation:

Indicates if this object has an active event state associated with it.

Property *eventState* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.eventState

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 36

Tag Value

An integer representing one of the options of enumeration **<BACnetEventState>**.

trendLogMultiple.eventTimeStamps

According to the BACnet protocol documentation:

Conveys the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events. Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

Property *eventTimeStamps* of object *trendLogMultiple* can be read and written using a block tag with 3 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.eventTimeStamps
Size	3 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 27
B3:	<objectAddress>
B4:	<property> = 130
Size:	3 elements

Block Elements

Index	Name	Type
0	elm00	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
1	elm01	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.
2	elm02	A Date, Integer, Null or String (Xml) value representing a timestamp in BACnet. • A null value means the timestamp is empty/not set; • A date/time value representing a precise date and time; • An integer defining the timestamp sequence number; • A character string containing an XML with a value of type <BACnetTimeStamp>.

trendLogMultiple.intervalOffset

According to the BACnet protocol documentation:

Specifies the offset in minutes from the beginning of the period specified for time synchronization until the actual time synchronization requests are sent.

Property *intervalOffset* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.intervalOffset

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 195

Tag Value

A 32-bit unsigned integer.

trendLogMultiple.lastNotifyRecord

According to the BACnet protocol documentation:

Represents the pPreviousCount parameter of the object's event algorithm.

Property *lastNotifyRecord* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.lastNotifyRecord

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 173

Tag Value

A 32-bit unsigned integer.

trendLogMultiple.logBuffer

According to the BACnet protocol documentation:

Contains a list of up to Buffer_Size timestamped log records of datatype BACnetEventLogRecord, each of which conveys the event notification parameters or status changes in the Event Log object.

Property *logBuffer* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.logBuffer

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 131

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetLogMultipleRecord>**.

trendLogMultiple.logDeviceObjectProperty

According to the BACnet protocol documentation:

Specifies the properties to be logged.

Property *logDeviceObjectProperty* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.logDeviceObjectProperty

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 132

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetDeviceObjectPropertyReference>**.

trendLogMultiple.logInterval

According to the BACnet protocol documentation:

Specifies the periodic interval in hundredths of seconds for which the referenced properties are to be logged when Logging_Type has the value POLLED. If Logging_Type has the value TRIGGERED, then the value of this property shall be zero and ignored. This property shall be writable if Logging_Type has the value POLLED, and shall be read-only if Logging_Type has the value TRIGGERED.

Property *logInterval* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.logInterval

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 134

Tag Value

A 32-bit unsigned integer.

trendLogMultiple.loggingType

According to the BACnet protocol documentation:

Specifies whether the Trend Log Multiple object collects log records using polling or triggered acquisition. COV Logging is not allowed for a Trend Log Multiple object.

Property *loggingType* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.loggingType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 197

Tag Value

An integer representing one of the options of enumeration **<BACnetLoggingType>**.

trendLogMultiple.notificationClass

According to the BACnet protocol documentation:

Specifies the notification class to be used when handling and generating event notifications for this object.

Property *notificationClass* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.notificationClass

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 17

Tag Value

A 32-bit unsigned integer.

trendLogMultiple.notificationThreshold

According to the BACnet protocol documentation:

Contains the pThreshold parameter for the object's event algorithm.

Property *notificationThreshold* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.notificationThreshold

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 137

Tag Value

A 32-bit unsigned integer.

trendLogMultiple.notifyType

According to the BACnet protocol documentation:

Conveys whether the notifications generated by the object should be Events (1) or Alarms (0).

Property *notifyType* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.notifyType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 72

Tag Value

An integer representing one of the options of enumeration **<BACnetNotifyType>**.

trendLogMultiple.objectIdentifier

According to the BACnet protocol documentation:

Numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.

Property *objectIdentifier* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.objectIdentifier

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 75

Tag Value

A string value containing a BACnet Object Identifier in the syntax '<Type>-<Address>' (ex: 'analogOutput-345').

trendLogMultiple.objectName

According to the BACnet protocol documentation:

String that represents a name for the object within the BACnet Device.

Property *objectName* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.objectName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 77

Tag Value

A string value.

trendLogMultiple.objectType

According to the BACnet protocol documentation:

Number indicating the object type.

Property *objectType* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.objectType

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 79

Tag Value

An integer representing one of the options of enumeration **<BACnetObjectType>**.

trendLogMultiple.profileLocation

According to the BACnet protocol documentation:

Contains the URI of the location of an xdd file containing the definition of the CSML type specified by the Profile_Name property and possible other information. The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

Property *profileLocation* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.profileLocation

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 485

Tag Value

A string value.

trendLogMultiple.profileName

According to the BACnet protocol documentation:

The name of an object profile to which this object conforms.

Property *profileName* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.profileName

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 168

Tag Value

A string value.

trendLogMultiple.propertyList

According to the BACnet protocol documentation:

Contains an array of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

Property *propertyList* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.propertyList

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 371

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetPropertyIdentifier>**.

trendLogMultiple.recordCount

According to the BACnet protocol documentation:

Represents the number of records currently resident in the log buffer.

Property *recordCount* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.recordCount

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 141

Tag Value

A 32-bit unsigned integer.

trendLogMultiple.recordsSinceNotification

According to the BACnet protocol documentation:

Represents the number of log records collected since the previous notification, or since the beginning of logging if no previous notification has occurred.

Property *recordsSinceNotification* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.recordsSinceNotification

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 140

Tag Value

A 32-bit unsigned integer.

trendLogMultiple.reliability

According to the BACnet protocol documentation:

Indicates weather the presentValue or the operation of the physical input in question is 'reliable' as far as the BACnet Device or operator can determine, and, if not, why.

Property *reliability* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.reliability

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 103

Tag Value

An integer representing one of the options of enumeration **<BACnetReliability>**.

trendLogMultiple.reliabilityEvaluationInhibit

According to the BACnet protocol documentation:

Indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation. When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

Property *reliabilityEvaluationInhibit* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.reliabilityEvaluationInhibit

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 357

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLogMultiple.startTime

According to the BACnet protocol documentation:

Specifies the date and time at or after which logging shall be enabled by this property. If this property contains an unspecified datetime, then the conditions for logging to be enabled by Start_Time shall be ignored. If Start_Time specifies a

date and time after Stop_Time, then logging shall be disabled. This property shall be writable if present. When Start_Time is reached, the value of the Enable property is not changed.

Property *startTime* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.startTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 142

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

trendLogMultiple.statusFlags

According to the BACnet protocol documentation:

Four boolean flags that indicate the general 'health' of an analog input.

Property *statusFlags* of object *trendLogMultiple* can be read and written using a block tag with 4 elements.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.statusFlags
Size	4 elements

Numeric Parameters

Property	Value
B1:	<deviceAddress>
B2:	<type> = 27
B3:	<objectAddress>
B4:	<property> = 111
Size:	4 elements

Block Elements

Index	Name	Type
0	inAlarm	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
1	fault	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
2	overridden	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.
3	outOfService	An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLogMultiple.stopTime

According to the BACnet protocol documentation:

Specifies the date and time at or after which logging shall be disabled by this property. If this property contains an unspecified datetime, then the conditions for logging to be disabled by Stop_Time shall be ignored. If Stop_Time specifies a date and time earlier than Start_Time, then logging shall be disabled. This property shall be writable if present. When Stop_Time is reached, the value of the Enable property is not changed.

Property *stopTime* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.stopTime

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 143

Tag Value

A Date, Null or String (Xml) value representing a BACnet date/time.

- A null value means the date/time is empty/not set;
- A date/time value representing a precise date and time;
- A character string containing an XML with a value of type **<BACnetDateTime>**.

trendLogMultiple.stopWhenFull

According to the BACnet protocol documentation:

Specifies whether (TRUE) or not (FALSE) logging should cease when the log buffer is full. When logging ceases because the addition of one more log record would cause the log buffer to be full, Enable shall be set to FALSE and the event recorded. If Stop_When_Full is writable, attempts to write the value TRUE to the Stop_When_Full property while Record_Count is equal to Buffer_Size shall result in the oldest log record in the log buffer being discarded, and shall cause the Enable property to be set to FALSE and the event to be recorded.

Property `stopWhenFull` of object `trendLogMultiple` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.stopWhenFull

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 144

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

trendLogMultiple.tags

According to the BACnet protocol documentation:

Contains a collection of tags for the object.

Property `tags` of object `trendLogMultiple` can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.tags

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 486

Tag Value

A character string containing an XML with a value of type **<SequenceOfBACnetNameValue>**.

trendLogMultiple.totalRecordCount

According to the BACnet protocol documentation:

Represents the total number of log records collected by the Trend Log object since creation. When the value of Total_Record_Count reaches its maximum possible value of 232 - 1, the next value it takes shall be one. Once this value has wrapped to one, its semantic value (the total number of log records collected) has been lost but its use in generating notifications remains.

Property *totalRecordCount* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.totalRecordCount

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 145

Tag Value

A 32-bit unsigned integer.

trendLogMultiple.trigger

According to the BACnet protocol documentation:

Causes the Trend Log object to acquire a log record whenever the value of this property is changed from FALSE to TRUE. It shall remain TRUE while the Trend Log object is acquiring the data items for a log record. When all data items have been collected or it has been determined that all outstanding data requests will not be fulfilled, the Trend Log object shall reset the value to FALSE.

Property *trigger* of object *trendLogMultiple* can be read and written using a single IOTag.

Text Parameters

Property	Value
ParamDevice	<deviceAddress>
ParamItem	trendLogMultiple-<objectAddress>.trigger

Numeric Parameters

Property	Value
N1:	<deviceAddress>
N2:	<type> = 27
N3:	<objectAddress>
N4:	<property> = 205

Tag Value

An integer value, where 0 corresponds to FALSE and any other value corresponds to TRUE.

XML data representation

The following BACnet objects are supported by the driver:

- Any
- BACnetAccessAuthenticationFactorDisable
- BACnetAccessCredentialDisable
- BACnetAccessCredentialDisableReason
- BACnetAccessEvent
- BACnetAccessRule
- BACnetAccessRule_LocationSpecifier
- BACnetAccessRule_TimeRangeSpecifier
- BACnetAccessZoneOccupancyState
- BACnetAction
- BACnetActionCommand
- BACnetActionList
- BACnetAddress
- BACnetAddressBinding
- BACnetAssignedAccessRights
- BACnetAssignedLandingCalls
- BACnetAssignedLandingCalls_LandingCalls
- BACnetAssignedLandingCalls_LandingCalls_Item
- BACnetAuthenticationFactor
- BACnetAuthenticationFactorFormat
- BACnetAuthenticationFactorType
- BACnetAuthenticationPolicy
- BACnetAuthenticationPolicy_Policy

- **BACnetAuthenticationPolicy_Policy_Item**
- **BACnetAuthenticationStatus**
- **BACnetAuthorizationExemption**
- **BACnetBDTEntry**
- **BACnetBackupState**
- **BACnetBinaryLightingPV**
- **BACnetBinaryPV**
- **BACnetCOVMultipleSubscription**
- **BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications**
- **BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item**
- **BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences**
- **BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences_Item**
- **BACnetCOVSubscription**
- **BACnetCalendarEntry**
- **BACnetCredentialAuthenticationFactor**
- **BACnetDailySchedule**
- **BACnetDateRange**
- **BACnetDateTime**
- **BACnetDaysOfWeek**
- **BACnetDestination**
- **BACnetDeviceObjectPropertyReference**
- **BACnetDeviceObjectPropertyValue**
- **BACnetDeviceObjectReference**
- **BACnetDeviceStatus**
- **BACnetDoorAlarmState**
- **BACnetDoorSecuredStatus**
- **BACnetDoorStatus**
- **BACnetDoorValue**
- **BACnetEngineeringUnits**
- **BACnetEscalatorFault**
- **BACnetEscalatorMode**
- **BACnetEscalatorOperationDirection**

- **BACnetEventLogRecord**
- **BACnetEventLogRecord_LogDatum**
- **BACnetEventNotificationSubscription**
- **BACnetEventParameter_AccessEvent**
- **BACnetEventParameter_BufferReady**
- **BACnetEventParameter_ChangeOfBitstring**
- **BACnetEventParameter_ChangeOfCharacterstring**
- **BACnetEventParameter_ChangeOfDiscreteValue**
- **BACnetEventParameter_ChangeOfLifeSafety**
- **BACnetEventParameter_ChangeOfState**
- **BACnetEventParameter_ChangeOfStatusFlags**
- **BACnetEventParameter_ChangeOfTimer**
- **BACnetEventParameter_ChangeOfValue**
- **BACnetEventParameter_ChangeOfValue_CovCriteria**
- **BACnetEventParameter_CommandFailure**
- **BACnetEventParameter_DoubleOutOfRange**
- **BACnetEventParameter_Extended**
- **BACnetEventParameter_Extended_Unnamed**
- **BACnetEventParameter_FloatingLimit**
- **BACnetEventParameter_OutOfRange**
- **BACnetEventParameter_SignedOutOfRange**
- **BACnetEventParameter_UnsignedOutOfRange**
- **BACnetEventParameter_UnsignedRange**
- **BACnetEventState**
- **BACnetEventTransitionBits**
- **BACnetEventType**
- **BACnetFDTEEntry**
- **BACnetFaultParameter_FaultCharacterstring**
- **BACnetFaultParameter_FaultExtended**
- **BACnetFaultParameter_FaultExtended_Unnamed**
- **BACnetFaultParameter_FaultLifeSafety**
- **BACnetFaultParameter_FaultListed**

- **BACnetFaultParameter_FaultOutOfRange**
- **BACnetFaultParameter_FaultOutOfRange_MaxNormalValue**
- **BACnetFaultParameter_FaultOutOfRange_MinNormalValue**
- **BACnetFaultParameter_FaultState**
- **BACnetFaultParameter_FaultStatusFlags**
- **BACnetFileAccessMethod**
- **BACnetHostAddress**
- **BACnetHostNPort**
- **BACnetIPMode**
- **BACnetKeyIdentifier**
- **BACnetLandingCallStatus**
- **BACnetLandingCallStatus_Command**
- **BACnetLandingDoorStatus**
- **BACnetLandingDoorStatus_LandingDoors**
- **BACnetLandingDoorStatus_LandingDoors_Item**
- **BACnetLifeSafetyMode**
- **BACnetLifeSafetyOperation**
- **BACnetLifeSafetyState**
- **BACnetLiftCarCallList**
- **BACnetLiftCarDirection**
- **BACnetLiftCarDoorCommand**
- **BACnetLiftCarDriveStatus**
- **BACnetLiftCarMode**
- **BACnetLiftFault**
- **BACnetLiftGroupMode**
- **BACnetLightingCommand**
- **BACnetLightingInProgress**
- **BACnetLightingOperation**
- **BACnetLightingTransition**
- **BACnetLockStatus**
- **BACnetLogData**
- **BACnetLogData_Unnamed**

- **BACnetLogMultipleRecord**
- **BACnetLogRecord**
- **BACnetLogRecord_LogDatum**
- **BACnetLogStatus**
- **BACnetMaintenance**
- **BACnetNameValue**
- **BACnetNameValueCollection**
- **BACnetNetworkNumberQuality**
- **BACnetNetworkPortCommand**
- **BACnetNetworkSecurityPolicy**
- **BACnetNetworkType**
- **BACnetNodeType**
- **BACnetNotificationParameters**
- **BACnetNotificationParameters_AccessEvent**
- **BACnetNotificationParameters_BufferReady**
- **BACnetNotificationParameters_ChangeOfBitstring**
- **BACnetNotificationParameters_ChangeOfCharacterstring**
- **BACnetNotificationParameters_ChangeOfDiscreteValue**
- **BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue**
- **BACnetNotificationParameters_ChangeOfLifeSafety**
- **BACnetNotificationParameters_ChangeOfReliability**
- **BACnetNotificationParameters_ChangeOfState**
- **BACnetNotificationParameters_ChangeOfStatusFlags**
- **BACnetNotificationParameters_ChangeOfTimer**
- **BACnetNotificationParameters_ChangeOfValue**
- **BACnetNotificationParameters_ChangeOfValue_NewValue**
- **BACnetNotificationParameters_CommandFailure**
- **BACnetNotificationParameters_DoubleOutOfRange**
- **BACnetNotificationParameters_Extended**
- **BACnetNotificationParameters_Extended_Unnamed**
- **BACnetNotificationParameters_FloatingLimit**
- **BACnetNotificationParameters_OutOfRange**

- **BACnetNotificationParameters_SignedOutOfRange**
- **BACnetNotificationParameters_UnsignedOutOfRange**
- **BACnetNotificationParameters_UnsignedRange**
- **BACnetNotifyType**
- **BACnetObjectIdentifier**
- **BACnetObjectPropertyReference**
- **BACnetOptionalCharacterString**
- **BACnetPolarity**
- **BACnetPortPermission**
- **BACnetProgramError**
- **BACnetProgramRequest**
- **BACnetProgramState**
- **BACnetPropertyAccessResult**
- **BACnetPropertyAccessResult_AccessResult**
- **BACnetPropertyIdentifier**
- **BACnetPropertyReference**
- **BACnetPropertyStates**
- **BACnetPropertyValue**
- **BACnetProtocolLevel**
- **BACnetRecipient**
- **BACnetRecipientProcess**
- **BACnetRelationship**
- **BACnetReliability**
- **BACnetRestartReason**
- **BACnetRouterEntry**
- **BACnetRouterEntry_Status**
- **BACnetSecurityKeySet**
- **BACnetSecurityLevel**
- **BACnetSecurityPolicy**
- **BACnetShedState**
- **BACnetSilencedState**
- **BACnetSpecialEvent**

- **BACnetSpecialEvent_Period**
- **BACnetStatusFlags**
- **BACnetTimeStamp**
- **BACnetTimeValue**
- **BACnetTimerState**
- **BACnetTimerTransition**
- **BACnetVMACEntry**
- **BACnetVTClass**
- **BACnetVTSession**
- **BACnetWriteStatus**
- **BitString**
- **Boolean**
- **CharacterString**
- **ConfirmedEventNotificationRequest**
- **Date**
- **Double**
- **Enumerated**
- **Error**
- **Error_ErrorClass**
- **Error_ErrorCode**
- **INTEGER**
- **Null**
- **OctetString**
- **REAL**
- **ReadAccessResult**
- **ReadAccessResult_ListOfResults**
- **ReadAccessResult_ListOfResults_Item**
- **ReadAccessResult_ListOfResults_Item_ReadResult**
- **ReadAccessSpecification**
- **SequenceOfBACnetAccessCredentialDisableReason**
- **SequenceOfBACnetAccessEvent**
- **SequenceOfBACnetAccessRule**

- **SequenceOfBACnetAccessZoneOccupancyState**
- **SequenceOfBACnetActionCommand**
- **SequenceOfBACnetActionList**
- **SequenceOfBACnetAddressBinding**
- **SequenceOfBACnetAssignedAccessRights**
- **SequenceOfBACnetAssignedLandingCalls**
- **SequenceOfBACnetAuthenticationFactorFormat**
- **SequenceOfBACnetAuthenticationPolicy**
- **SequenceOfBACnetAuthorizationExemption**
- **SequenceOfBACnetBDTEntry**
- **SequenceOfBACnetCOVMultipleSubscription**
- **SequenceOfBACnetCOVSubscription**
- **SequenceOfBACnetCalendarEntry**
- **SequenceOfBACnetCredentialAuthenticationFactor**
- **SequenceOfBACnetDestination**
- **SequenceOfBACnetDeviceObjectPropertyReference**
- **SequenceOfBACnetDeviceObjectReference**
- **SequenceOfBACnetDoorAlarmState**
- **SequenceOfBACnetDoorStatus**
- **SequenceOfBACnetEscalatorFault**
- **SequenceOfBACnetEventLogRecord**
- **SequenceOfBACnetEventNotificationSubscription**
- **SequenceOfBACnetEventParameter_Extended_Unnamed**
- **SequenceOfBACnetFDTEEntry**
- **SequenceOfBACnetFaultParameter_FaultExtended_Unnamed**
- **SequenceOfBACnetKeyIdentifier**
- **SequenceOfBACnetLandingCallStatus**
- **SequenceOfBACnetLandingDoorStatus**
- **SequenceOfBACnetLifeSafetyMode**
- **SequenceOfBACnetLifeSafetyState**
- **SequenceOfBACnetLiftCarCallList**
- **SequenceOfBACnetLiftCarDoorCommand**

- **SequenceOfBACnetLiftFault**
- **SequenceOfBACnetLogData_Unnamed**
- **SequenceOfBACnetLogMultipleRecord**
- **SequenceOfBACnetLogRecord**
- **SequenceOfBACnetNameValue**
- **SequenceOfBACnetNameValueCollection**
- **SequenceOfBACnetNetworkSecurityPolicy**
- **SequenceOfBACnetNodeType**
- **SequenceOfBACnetNotificationParameters_Extended_Unnamed**
- **SequenceOfBACnetObjectIdentifier**
- **SequenceOfBACnetOptionalCharacterString**
- **SequenceOfBACnetPortPermission**
- **SequenceOfBACnetPropertyAccessResult**
- **SequenceOfBACnetPropertyIdentifier**
- **SequenceOfBACnetPropertyReference**
- **SequenceOfBACnetPropertyStates**
- **SequenceOfBACnetPropertyValue**
- **SequenceOfBACnetRecipient**
- **SequenceOfBACnetRelationship**
- **SequenceOfBACnetRouterEntry**
- **SequenceOfBACnetSpecialEvent**
- **SequenceOfBACnetTimeValue**
- **SequenceOfBACnetTimerState**
- **SequenceOfBACnetVMACEntry**
- **SequenceOfBACnetVTClass**
- **SequenceOfBACnetVTSession**
- **SequenceOfBitString**
- **SequenceOfCharacterString**
- **SequenceOfOctetString**
- **SequenceOfREAL**
- **SequenceOfReadAccessResult**
- **SequenceOfReadAccessSpecification**

- **SequenceOfUnsigned**
- **SequenceOfUnsigned32**
- **SequenceOfUnsigned8**
- **Time**
- **Unsigned**
- **Unsigned16**
- **Unsigned32**
- **Unsigned8**

Any

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.priorityArray**
- **analogOutput.priorityArray**
- **analogValue.priorityArray**
- **binaryLightingOutput.priorityArray**
- **binaryOutput.priorityArray**
- **binaryValue.priorityArray**
- **bitstringValue.priorityArray**
- **characterstringValue.priorityArray**
- **dateValue.priorityArray**
- **datepatternValue.priorityArray**
- **datetimeValue.priorityArray**
- **datetimepatternValue.priorityArray**
- **integerValue.priorityArray**
- **largeAnalogValue.priorityArray**
- **lightingOutput.priorityArray**
- **multiStateOutput.priorityArray**
- **multiStateValue.priorityArray**
- **octetstringValue.priorityArray**
- **positiveIntegerValue.priorityArray**
- **timeValue.priorityArray**
- **timepatternValue.priorityArray**
- **timer.stateChangeValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="Any">
    <xs:sequence>
      <xs:choice>
        <xs:element name="BACnetAccessAuthenticationFactorDisable"
type="BACnetAccessAuthenticationFactorDisable" />
        <xs:element name="BACnetAccessCredentialDisable" type="BACnetAccessCredentialDisable" />
        <xs:element name="BACnetAccessCredentialDisableReason"
type="BACnetAccessCredentialDisableReason" />
        <xs:element name="BACnetAccessEvent" type="BACnetAccessEvent" />
        <xs:element name="BACnetAccessPassbackMode" type="BACnetAccessPassbackMode" />
        <xs:element name="BACnetAccessRule" type="BACnetAccessRule" />
        <xs:element name="BACnetAccessRule_LocationSpecifier"
type="BACnetAccessRule_LocationSpecifier" />
        <xs:element name="BACnetAccessRule_TimeRangeSpecifier"
type="BACnetAccessRule_TimeRangeSpecifier" />
        <xs:element name="BACnetAccessUserType" type="BACnetAccessUserType" />
        <xs:element name="BACnetAccessZoneOccupancyState" type="BACnetAccessZoneOccupancyState" />
        <xs:element name="BACnetAccumulatorRecord" type="BACnetAccumulatorRecord" />
        <xs:element name="BACnetAccumulatorRecord_AccumulatorStatus"
type="BACnetAccumulatorRecord_AccumulatorStatus" />
        <xs:element name="BACnetAction" type="BACnetAction" />
        <xs:element name="BACnetActionCommand" type="BACnetActionCommand" />
        <xs:element name="BACnetActionList" type="BACnetActionList" />
        <xs:element name="BACnetAddress" type="BACnetAddress" />
        <xs:element name="BACnetAddressBinding" type="BACnetAddressBinding" />
        <xs:element name="BACnetAssignedAccessRights" type="BACnetAssignedAccessRights" />
        <xs:element name="BACnetAssignedLandingCalls" type="BACnetAssignedLandingCalls" />
        <xs:element name="BACnetAssignedLandingCalls_LandingCalls"
type="BACnetAssignedLandingCalls_LandingCalls" />
        <xs:element name="BACnetAssignedLandingCalls_LandingCalls_Item"
type="BACnetAssignedLandingCalls_LandingCalls_Item" />
        <xs:element name="BACnetAuthenticationFactor" type="BACnetAuthenticationFactor" />
        <xs:element name="BACnetAuthenticationFactorFormat"
type="BACnetAuthenticationFactorFormat" />
        <xs:element name="BACnetAuthenticationFactorType" type="BACnetAuthenticationFactorType" />
        <xs:element name="BACnetAuthenticationPolicy" type="BACnetAuthenticationPolicy" />
        <xs:element name="BACnetAuthenticationPolicy_Policy"
type="BACnetAuthenticationPolicy_Policy" />
        <xs:element name="BACnetAuthenticationPolicy_Policy_Item"
type="BACnetAuthenticationPolicy_Policy_Item" />
        <xs:element name="BACnetAuthenticationStatus" type="BACnetAuthenticationStatus" />
        <xs:element name="BACnetAuthorizationExemption" type="BACnetAuthorizationExemption" />
        <xs:element name="BACnetAuthorizationMode" type="BACnetAuthorizationMode" />
        <xs:element name="BACnetBackupState" type="BACnetBackupState" />
        <xs:element name="BACnetBDTEntry" type="BACnetBDTEntry" />
        <xs:element name="BACnetBinaryLightingPV" type="BACnetBinaryLightingPV" />
        <xs:element name="BACnetBinaryPV" type="BACnetBinaryPV" />
        <xs:element name="BACnetCalendarEntry" type="BACnetCalendarEntry" />
        <xs:element name="BACnetChannelValue" type="BACnetChannelValue" />
        <xs:element name="BACnetClientCOV" type="BACnetClientCOV" />
        <xs:element name="BACnetCOVMultipleSubscription" type="BACnetCOVMultipleSubscription" />
        <xs:element name="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications"
type="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications" />
        <xs:element name="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item"
type="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item" />
        <xs:element
name="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences"
type="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences" />
        <xs:element
name="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences_Ite
m"
type="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences_Ite
m" />
        <xs:element name="BACnetCOVSubscription" type="BACnetCOVSubscription" />
        <xs:element name="BACnetCredentialAuthenticationFactor"
type="BACnetCredentialAuthenticationFactor" />

```

```

        <xs:element name="BACnetDailySchedule" type="BACnetDailySchedule" />
        <xs:element name="BACnetDateRange" type="BACnetDateRange" />
        <xs:element name="BACnetDateTime" type="BACnetDateTime" />
        <xs:element name="BACnetDaysOfWeek" type="BACnetDaysOfWeek" />
        <xs:element name="BACnetDestination" type="BACnetDestination" />
        <xs:element name="BACnetDeviceObjectPropertyReference"
type="BACnetDeviceObjectPropertyReference" />
        <xs:element name="BACnetDeviceObjectPropertyValue" type="BACnetDeviceObjectPropertyValue" />
        <xs:element name="BACnetDeviceObjectReference" type="BACnetDeviceObjectReference" />
        <xs:element name="BACnetDeviceStatus" type="BACnetDeviceStatus" />
        <xs:element name="BACnetDoorAlarmState" type="BACnetDoorAlarmState" />
        <xs:element name="BACnetDoorSecuredStatus" type="BACnetDoorSecuredStatus" />
        <xs:element name="BACnetDoorStatus" type="BACnetDoorStatus" />
        <xs:element name="BACnetDoorValue" type="BACnetDoorValue" />
        <xs:element name="BACnetEngineeringUnits" type="BACnetEngineeringUnits" />
        <xs:element name="BACnetEscalatorFault" type="BACnetEscalatorFault" />
        <xs:element name="BACnetEscalatorMode" type="BACnetEscalatorMode" />
        <xs:element name="BACnetEscalatorOperationDirection"
type="BACnetEscalatorOperationDirection" />
        <xs:element name="BACnetEventLogRecord" type="BACnetEventLogRecord" />
        <xs:element name="BACnetEventLogRecord_LogDatum" type="BACnetEventLogRecord_LogDatum" />
        <xs:element name="BACnetEventNotificationSubscription"
type="BACnetEventNotificationSubscription" />
        <xs:element name="BACnetEventParameter" type="BACnetEventParameter" />
        <xs:element name="BACnetEventParameter_AccessEvent"
type="BACnetEventParameter_AccessEvent" />
        <xs:element name="BACnetEventParameter_BufferReady"
type="BACnetEventParameter_BufferReady" />
        <xs:element name="BACnetEventParameter_ChangeOfBitstring"
type="BACnetEventParameter_ChangeOfBitstring" />
        <xs:element name="BACnetEventParameter_ChangeOfCharacterstring"
type="BACnetEventParameter_ChangeOfCharacterstring" />
        <xs:element name="BACnetEventParameter_ChangeOfDiscreteValue"
type="BACnetEventParameter_ChangeOfDiscreteValue" />
        <xs:element name="BACnetEventParameter_ChangeOfLifeSafety"
type="BACnetEventParameter_ChangeOfLifeSafety" />
        <xs:element name="BACnetEventParameter_ChangeOfState"
type="BACnetEventParameter_ChangeOfState" />
        <xs:element name="BACnetEventParameter_ChangeOfStatusFlags"
type="BACnetEventParameter_ChangeOfStatusFlags" />
        <xs:element name="BACnetEventParameter_ChangeOfTimer"
type="BACnetEventParameter_ChangeOfTimer" />
        <xs:element name="BACnetEventParameter_ChangeOfValue"
type="BACnetEventParameter_ChangeOfValue" />
        <xs:element name="BACnetEventParameter_ChangeOfValue_CovCriteria"
type="BACnetEventParameter_ChangeOfValue_CovCriteria" />
        <xs:element name="BACnetEventParameter_CommandFailure"
type="BACnetEventParameter_CommandFailure" />
        <xs:element name="BACnetEventParameter_DoubleOutOfRange"
type="BACnetEventParameter_DoubleOutOfRange" />
        <xs:element name="BACnetEventParameter_Extended" type="BACnetEventParameter_Extended" />
        <xs:element name="BACnetEventParameter_Extended_Unnamed"
type="BACnetEventParameter_Extended_Unnamed" />
        <xs:element name="BACnetEventParameter_FloatingLimit"
type="BACnetEventParameter_FloatingLimit" />
        <xs:element name="BACnetEventParameter_OutOfRange" type="BACnetEventParameter_OutOfRange" />
        <xs:element name="BACnetEventParameter_SignedOutOfRange"
type="BACnetEventParameter_SignedOutOfRange" />
        <xs:element name="BACnetEventParameter_UnsignedOutOfRange"
type="BACnetEventParameter_UnsignedOutOfRange" />
        <xs:element name="BACnetEventParameter_UnsignedRange"
type="BACnetEventParameter_UnsignedRange" />
        <xs:element name="BACnetEventState" type="BACnetEventState" />
        <xs:element name="BACnetEventTransitionBits" type="BACnetEventTransitionBits" />
        <xs:element name="BACnetEventType" type="BACnetEventType" />
        <xs:element name="BACnetFaultParameter" type="BACnetFaultParameter" />
        <xs:element name="BACnetFaultParameter_FaultCharacterstring"
type="BACnetFaultParameter_FaultCharacterstring" />
        <xs:element name="BACnetFaultParameter_FaultExtended"
type="BACnetFaultParameter_FaultExtended" />
        <xs:element name="BACnetFaultParameter_FaultExtended_Unnamed"

```

```

type="BACnetFaultParameter_FaultExtended_Unnamed" />
  <xs:element name="BACnetFaultParameter_FaultLifeSafety"
type="BACnetFaultParameter_FaultLifeSafety" />
  <xs:element name="BACnetFaultParameter_FaultListed"
type="BACnetFaultParameter_FaultListed" />
  <xs:element name="BACnetFaultParameter_FaultOutOfRange"
type="BACnetFaultParameter_FaultOutOfRange" />
  <xs:element name="BACnetFaultParameter_FaultOutOfRange_MaxNormalValue"
type="BACnetFaultParameter_FaultOutOfRange_MaxNormalValue" />
  <xs:element name="BACnetFaultParameter_FaultOutOfRange_MinNormalValue"
type="BACnetFaultParameter_FaultOutOfRange_MinNormalValue" />
  <xs:element name="BACnetFaultParameter_FaultState" type="BACnetFaultParameter_FaultState" />
  <xs:element name="BACnetFaultParameter_FaultStatusFlags"
type="BACnetFaultParameter_FaultStatusFlags" />
  <xs:element name="BACnetFaultType" type="BACnetFaultType" />
  <xs:element name="BACnetFDTEEntry" type="BACnetFDTEEntry" />
  <xs:element name="BACnetFileAccessMethod" type="BACnetFileAccessMethod" />
  <xs:element name="BACnetHostAddress" type="BACnetHostAddress" />
  <xs:element name="BACnetHostNPort" type="BACnetHostNPort" />
  <xs:element name="BACnetIPMode" type="BACnetIPMode" />
  <xs:element name="BACnetKeyIdentifier" type="BACnetKeyIdentifier" />
  <xs:element name="BACnetLandingCallStatus" type="BACnetLandingCallStatus" />
  <xs:element name="BACnetLandingCallStatus_Command" type="BACnetLandingCallStatus_Command" />
  <xs:element name="BACnetLandingDoorStatus" type="BACnetLandingDoorStatus" />
  <xs:element name="BACnetLandingDoorStatus_LandingDoors"
type="BACnetLandingDoorStatus_LandingDoors" />
  <xs:element name="BACnetLandingDoorStatus_LandingDoors_Item"
type="BACnetLandingDoorStatus_LandingDoors_Item" />
  <xs:element name="BACnetLifeSafetyMode" type="BACnetLifeSafetyMode" />
  <xs:element name="BACnetLifeSafetyOperation" type="BACnetLifeSafetyOperation" />
  <xs:element name="BACnetLifeSafetyState" type="BACnetLifeSafetyState" />
  <xs:element name="BACnetLiftCarCallList" type="BACnetLiftCarCallList" />
  <xs:element name="BACnetLiftCarDirection" type="BACnetLiftCarDirection" />
  <xs:element name="BACnetLiftCarDoorCommand" type="BACnetLiftCarDoorCommand" />
  <xs:element name="BACnetLiftCarDriveStatus" type="BACnetLiftCarDriveStatus" />
  <xs:element name="BACnetLiftCarMode" type="BACnetLiftCarMode" />
  <xs:element name="BACnetLiftFault" type="BACnetLiftFault" />
  <xs:element name="BACnetLiftGroupMode" type="BACnetLiftGroupMode" />
  <xs:element name="BACnetLightingCommand" type="BACnetLightingCommand" />
  <xs:element name="BACnetLightingInProgress" type="BACnetLightingInProgress" />
  <xs:element name="BACnetLightingOperation" type="BACnetLightingOperation" />
  <xs:element name="BACnetLightingTransition" type="BACnetLightingTransition" />
  <xs:element name="BACnetLimitEnable" type="BACnetLimitEnable" />
  <xs:element name="BACnetLockStatus" type="BACnetLockStatus" />
  <xs:element name="BACnetLogData" type="BACnetLogData" />
  <xs:element name="BACnetLogData_Unnamed" type="BACnetLogData_Unnamed" />
  <xs:element name="BACnetLoggingType" type="BACnetLoggingType" />
  <xs:element name="BACnetLogMultipleRecord" type="BACnetLogMultipleRecord" />
  <xs:element name="BACnetLogRecord" type="BACnetLogRecord" />
  <xs:element name="BACnetLogRecord_LogDatum" type="BACnetLogRecord_LogDatum" />
  <xs:element name="BACnetLogStatus" type="BACnetLogStatus" />
  <xs:element name="BACnetMaintenance" type="BACnetMaintenance" />
  <xs:element name="BACnetNameValue" type="BACnetNameValue" />
  <xs:element name="BACnetNameValueCollection" type="BACnetNameValueCollection" />
  <xs:element name="BACnetNetworkNumberQuality" type="BACnetNetworkNumberQuality" />
  <xs:element name="BACnetNetworkPortCommand" type="BACnetNetworkPortCommand" />
  <xs:element name="BACnetNetworkSecurityPolicy" type="BACnetNetworkSecurityPolicy" />
  <xs:element name="BACnetNetworkType" type="BACnetNetworkType" />
  <xs:element name="BACnetNodeType" type="BACnetNodeType" />
  <xs:element name="BACnetNotificationParameters" type="BACnetNotificationParameters" />
  <xs:element name="BACnetNotificationParameters_AccessEvent"
type="BACnetNotificationParameters_AccessEvent" />
  <xs:element name="BACnetNotificationParameters_BufferReady"
type="BACnetNotificationParameters_BufferReady" />
  <xs:element name="BACnetNotificationParameters_ChangeOfBitstring"
type="BACnetNotificationParameters_ChangeOfBitstring" />
  <xs:element name="BACnetNotificationParameters_ChangeOfCharacterstring"
type="BACnetNotificationParameters_ChangeOfCharacterstring" />
  <xs:element name="BACnetNotificationParameters_ChangeOfDiscreteValue"
type="BACnetNotificationParameters_ChangeOfDiscreteValue" />
  <xs:element name="BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue"

```

```

type="BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue" />
  <xs:element name="BACnetNotificationParameters_ChangeOfLifeSafety"
type="BACnetNotificationParameters_ChangeOfLifeSafety" />
  <xs:element name="BACnetNotificationParameters_ChangeOfReliability"
type="BACnetNotificationParameters_ChangeOfReliability" />
  <xs:element name="BACnetNotificationParameters_ChangeOfState"
type="BACnetNotificationParameters_ChangeOfState" />
  <xs:element name="BACnetNotificationParameters_ChangeOfStatusFlags"
type="BACnetNotificationParameters_ChangeOfStatusFlags" />
  <xs:element name="BACnetNotificationParameters_ChangeOfTimer"
type="BACnetNotificationParameters_ChangeOfTimer" />
  <xs:element name="BACnetNotificationParameters_ChangeOfValue"
type="BACnetNotificationParameters_ChangeOfValue" />
  <xs:element name="BACnetNotificationParameters_ChangeOfValue_NewValue"
type="BACnetNotificationParameters_ChangeOfValue_NewValue" />
  <xs:element name="BACnetNotificationParameters_CommandFailure"
type="BACnetNotificationParameters_CommandFailure" />
  <xs:element name="BACnetNotificationParameters_DoubleOutOfRange"
type="BACnetNotificationParameters_DoubleOutOfRange" />
  <xs:element name="BACnetNotificationParameters_Extended"
type="BACnetNotificationParameters_Extended" />
  <xs:element name="BACnetNotificationParameters_Extended_Unnamed"
type="BACnetNotificationParameters_Extended_Unnamed" />
  <xs:element name="BACnetNotificationParameters_FloatingLimit"
type="BACnetNotificationParameters_FloatingLimit" />
  <xs:element name="BACnetNotificationParameters_OutOfRange"
type="BACnetNotificationParameters_OutOfRange" />
  <xs:element name="BACnetNotificationParameters_SignedOutOfRange"
type="BACnetNotificationParameters_SignedOutOfRange" />
  <xs:element name="BACnetNotificationParameters_UnsignedOutOfRange"
type="BACnetNotificationParameters_UnsignedOutOfRange" />
  <xs:element name="BACnetNotificationParameters_UnsignedRange"
type="BACnetNotificationParameters_UnsignedRange" />
  <xs:element name="BACnetNotifyType" type="BACnetNotifyType" />
  <xs:element name="BACnetObjectIdentifier" type="BACnetObjectIdentifier" />
  <xs:element name="BACnetObjectPropertyReference" type="BACnetObjectPropertyReference" />
  <xs:element name="BACnetObjectType" type="BACnetObjectType" />
  <xs:element name="BACnetObjectTypesSupported" type="BACnetObjectTypesSupported" />
  <xs:element name="BACnetOptionalBinaryPV" type="BACnetOptionalBinaryPV" />
  <xs:element name="BACnetOptionalCharacterString" type="BACnetOptionalCharacterString" />
  <xs:element name="BACnetOptionalREAL" type="BACnetOptionalREAL" />
  <xs:element name="BACnetOptionalUnsigned" type="BACnetOptionalUnsigned" />
  <xs:element name="BACnetPolarity" type="BACnetPolarity" />
  <xs:element name="BACnetPortPermission" type="BACnetPortPermission" />
  <xs:element name="BACnetPrescale" type="BACnetPrescale" />
  <xs:element name="BACnetPriorityArray" type="BACnetPriorityArray" />
  <xs:element name="BACnetPriorityValue" type="BACnetPriorityValue" />
  <xs:element name="BACnetProcessIdSelection" type="BACnetProcessIdSelection" />
  <xs:element name="BACnetProgramError" type="BACnetProgramError" />
  <xs:element name="BACnetProgramRequest" type="BACnetProgramRequest" />
  <xs:element name="BACnetProgramState" type="BACnetProgramState" />
  <xs:element name="BACnetPropertyAccessResult" type="BACnetPropertyAccessResult" />
  <xs:element name="BACnetPropertyAccessResult_AccessResult"
type="BACnetPropertyAccessResult_AccessResult" />
  <xs:element name="BACnetPropertyIdentifier" type="BACnetPropertyIdentifier" />
  <xs:element name="BACnetPropertyReference" type="BACnetPropertyReference" />
  <xs:element name="BACnetPropertyStates" type="BACnetPropertyStates" />
  <xs:element name="BACnetPropertyValue" type="BACnetPropertyValue" />
  <xs:element name="BACnetProtocolLevel" type="BACnetProtocolLevel" />
  <xs:element name="BACnetRecipient" type="BACnetRecipient" />
  <xs:element name="BACnetRecipientProcess" type="BACnetRecipientProcess" />
  <xs:element name="BACnetRelationship" type="BACnetRelationship" />
  <xs:element name="BACnetReliability" type="BACnetReliability" />
  <xs:element name="BACnetRestartReason" type="BACnetRestartReason" />
  <xs:element name="BACnetRouterEntry" type="BACnetRouterEntry" />
  <xs:element name="BACnetRouterEntry_Status" type="BACnetRouterEntry_Status" />
  <xs:element name="BACnetScale" type="BACnetScale" />
  <xs:element name="BACnetSecurityKeySet" type="BACnetSecurityKeySet" />
  <xs:element name="BACnetSecurityLevel" type="BACnetSecurityLevel" />
  <xs:element name="BACnetSecurityPolicy" type="BACnetSecurityPolicy" />
  <xs:element name="BACnetSegmentation" type="BACnetSegmentation" />

```



```

<xs:element name="BACnetServicesSupported" type="BACnetServicesSupported" />
<xs:element name="BACnetSetpointReference" type="BACnetSetpointReference" />
<xs:element name="BACnetShedLevel" type="BACnetShedLevel" />
<xs:element name="BACnetShedState" type="BACnetShedState" />
<xs:element name="BACnetSilencedState" type="BACnetSilencedState" />
<xs:element name="BACnetSpecialEvent" type="BACnetSpecialEvent" />
<xs:element name="BACnetSpecialEvent_Period" type="BACnetSpecialEvent_Period" />
<xs:element name="BACnetStatusFlags" type="BACnetStatusFlags" />
<xs:element name="BACnetTimerState" type="BACnetTimerState" />
<xs:element name="BACnetTimerStateChangeValue" type="BACnetTimerStateChangeValue" />
<xs:element name="BACnetTimerTransition" type="BACnetTimerTransition" />
<xs:element name="BACnetTimeStamp" type="BACnetTimeStamp" />
<xs:element name="BACnetTimeValue" type="BACnetTimeValue" />
<xs:element name="BACnetValueSource" type="BACnetValueSource" />
<xs:element name="BACnetVMACEntry" type="BACnetVMACEntry" />
<xs:element name="BACnetVTClass" type="BACnetVTClass" />
<xs:element name="BACnetVTSession" type="BACnetVTSession" />
<xs:element name="BACnetWeekNDay" type="BACnetWeekNDay" />
<xs:element name="BACnetWriteStatus" type="BACnetWriteStatus" />
<xs:element name="BitString" type="BitString" />
<xs:element name="Boolean" type="Boolean" />
<xs:element name="CharacterString" type="CharacterString" />
<xs:element name="ConfirmedEventNotificationRequest"
type="ConfirmedEventNotificationRequest" />
<xs:element name="Date" type="Date" />
<xs:element name="Double" type="Double" />
<xs:element name="Enumerated" type="Enumerated" />
<xs:element name="Error" type="Error" />
<xs:element name="Error_ErrorClass" type="Error_ErrorClass" />
<xs:element name="Error_ErrorCode" type="Error_ErrorCode" />
<xs:element name="INTEGER" type="INTEGER" />
<xs:element name="Null" type="Null" />
<xs:element name="OctetString" type="OctetString" />
<xs:element name="ReadAccessResult" type="ReadAccessResult" />
<xs:element name="ReadAccessResult_ListOfResults" type="ReadAccessResult_ListOfResults" />
<xs:element name="ReadAccessResult_ListOfResults_Item"
type="ReadAccessResult_ListOfResults_Item" />
<xs:element name="ReadAccessResult_ListOfResults_Item_ReadResult"
type="ReadAccessResult_ListOfResults_Item_ReadResult" />
<xs:element name="ReadAccessSpecification" type="ReadAccessSpecification" />
<xs:element name="REAL" type="REAL" />
<xs:element name="SequenceOf16BACnetTimeStamp" type="SequenceOf16BACnetTimeStamp" />
<xs:element name="SequenceOf16BACnetValueSource" type="SequenceOf16BACnetValueSource" />
<xs:element name="SequenceOf2BACnetSecurityKeySet" type="SequenceOf2BACnetSecurityKeySet" />
<xs:element name="SequenceOf3BACnetTimeStamp" type="SequenceOf3BACnetTimeStamp" />
<xs:element name="SequenceOf3CharacterString" type="SequenceOf3CharacterString" />
<xs:element name="SequenceOf3Unsigned" type="SequenceOf3Unsigned" />
<xs:element name="SequenceOf7BACnetDailySchedule" type="SequenceOf7BACnetDailySchedule" />
<xs:element name="SequenceOf7BACnetTimerStateChangeValue"
type="SequenceOf7BACnetTimerStateChangeValue" />
<xs:element name="SequenceOfBACnetAccessCredentialDisableReason"
type="SequenceOfBACnetAccessCredentialDisableReason" />
<xs:element name="SequenceOfBACnetAccessEvent" type="SequenceOfBACnetAccessEvent" />
<xs:element name="SequenceOfBACnetAccessRule" type="SequenceOfBACnetAccessRule" />
<xs:element name="SequenceOfBACnetAccessZoneOccupancyState"
type="SequenceOfBACnetAccessZoneOccupancyState" />
<xs:element name="SequenceOfBACnetActionCommand" type="SequenceOfBACnetActionCommand" />
<xs:element name="SequenceOfBACnetActionList" type="SequenceOfBACnetActionList" />
<xs:element name="SequenceOfBACnetAddressBinding" type="SequenceOfBACnetAddressBinding" />
<xs:element name="SequenceOfBACnetAssignedAccessRights"
type="SequenceOfBACnetAssignedAccessRights" />
<xs:element name="SequenceOfBACnetAssignedLandingCalls"
type="SequenceOfBACnetAssignedLandingCalls" />
<xs:element name="SequenceOfBACnetAuthenticationFactorFormat"
type="SequenceOfBACnetAuthenticationFactorFormat" />
<xs:element name="SequenceOfBACnetAuthenticationPolicy"
type="SequenceOfBACnetAuthenticationPolicy" />
<xs:element name="SequenceOfBACnetAuthorizationExemption"
type="SequenceOfBACnetAuthorizationExemption" />
<xs:element name="SequenceOfBACnetBDTEntry" type="SequenceOfBACnetBDTEntry" />
<xs:element name="SequenceOfBACnetCalendarEntry" type="SequenceOfBACnetCalendarEntry" />

```

```

        <xs:element name="SequenceOfBACnetCOVMultipleSubscription"
type="SequenceOfBACnetCOVMultipleSubscription" />
        <xs:element name="SequenceOfBACnetCOVSubscription" type="SequenceOfBACnetCOVSubscription" />
        <xs:element name="SequenceOfBACnetCredentialAuthenticationFactor"
type="SequenceOfBACnetCredentialAuthenticationFactor" />
        <xs:element name="SequenceOfBACnetDestination" type="SequenceOfBACnetDestination" />
        <xs:element name="SequenceOfBACnetDeviceObjectPropertyReference"
type="SequenceOfBACnetDeviceObjectPropertyReference" />
        <xs:element name="SequenceOfBACnetDeviceObjectReference"
type="SequenceOfBACnetDeviceObjectReference" />
        <xs:element name="SequenceOfBACnetDoorAlarmState" type="SequenceOfBACnetDoorAlarmState" />
        <xs:element name="SequenceOfBACnetDoorStatus" type="SequenceOfBACnetDoorStatus" />
        <xs:element name="SequenceOfBACnetEscalatorFault" type="SequenceOfBACnetEscalatorFault" />
        <xs:element name="SequenceOfBACnetEventLogRecord" type="SequenceOfBACnetEventLogRecord" />
        <xs:element name="SequenceOfBACnetEventNotificationSubscription"
type="SequenceOfBACnetEventNotificationSubscription" />
        <xs:element name="SequenceOfBACnetEventParameter_Extended_Unnamed"
type="SequenceOfBACnetEventParameter_Extended_Unnamed" />
        <xs:element name="SequenceOfBACnetFaultParameter_FaultExtended_Unnamed"
type="SequenceOfBACnetFaultParameter_FaultExtended_Unnamed" />
        <xs:element name="SequenceOfBACnetFDTEEntry" type="SequenceOfBACnetFDTEEntry" />
        <xs:element name="SequenceOfBACnetKeyIdentifier" type="SequenceOfBACnetKeyIdentifier" />
        <xs:element name="SequenceOfBACnetLandingCallStatus"
type="SequenceOfBACnetLandingCallStatus" />
        <xs:element name="SequenceOfBACnetLandingDoorStatus"
type="SequenceOfBACnetLandingDoorStatus" />
        <xs:element name="SequenceOfBACnetLifeSafetyMode" type="SequenceOfBACnetLifeSafetyMode" />
        <xs:element name="SequenceOfBACnetLifeSafetyState" type="SequenceOfBACnetLifeSafetyState" />
        <xs:element name="SequenceOfBACnetLiftCarCallList" type="SequenceOfBACnetLiftCarCallList" />
        <xs:element name="SequenceOfBACnetLiftCarDoorCommand"
type="SequenceOfBACnetLiftCarDoorCommand" />
        <xs:element name="SequenceOfBACnetLiftFault" type="SequenceOfBACnetLiftFault" />
        <xs:element name="SequenceOfBACnetLogData_Unnamed" type="SequenceOfBACnetLogData_Unnamed" />
        <xs:element name="SequenceOfBACnetLogMultipleRecord"
type="SequenceOfBACnetLogMultipleRecord" />
        <xs:element name="SequenceOfBACnetLogRecord" type="SequenceOfBACnetLogRecord" />
        <xs:element name="SequenceOfBACnetNameValue" type="SequenceOfBACnetNameValue" />
        <xs:element name="SequenceOfBACnetNameValueCollection"
type="SequenceOfBACnetNameValueCollection" />
        <xs:element name="SequenceOfBACnetNetworkSecurityPolicy"
type="SequenceOfBACnetNetworkSecurityPolicy" />
        <xs:element name="SequenceOfBACnetNodeType" type="SequenceOfBACnetNodeType" />
        <xs:element name="SequenceOfBACnetNotificationParameters_Extended_Unnamed"
type="SequenceOfBACnetNotificationParameters_Extended_Unnamed" />
        <xs:element name="SequenceOfBACnetObjectIdentifier"
type="SequenceOfBACnetObjectIdentifier" />
        <xs:element name="SequenceOfBACnetOptionalCharacterString"
type="SequenceOfBACnetOptionalCharacterString" />
        <xs:element name="SequenceOfBACnetPortPermission" type="SequenceOfBACnetPortPermission" />
        <xs:element name="SequenceOfBACnetPropertyAccessResult"
type="SequenceOfBACnetPropertyAccessResult" />
        <xs:element name="SequenceOfBACnetPropertyIdentifier"
type="SequenceOfBACnetPropertyIdentifier" />
        <xs:element name="SequenceOfBACnetPropertyReference"
type="SequenceOfBACnetPropertyReference" />
        <xs:element name="SequenceOfBACnetPropertyStates" type="SequenceOfBACnetPropertyStates" />
        <xs:element name="SequenceOfBACnetPropertyValue" type="SequenceOfBACnetPropertyValue" />
        <xs:element name="SequenceOfBACnetRecipient" type="SequenceOfBACnetRecipient" />
        <xs:element name="SequenceOfBACnetRelationship" type="SequenceOfBACnetRelationship" />
        <xs:element name="SequenceOfBACnetRouterEntry" type="SequenceOfBACnetRouterEntry" />
        <xs:element name="SequenceOfBACnetSpecialEvent" type="SequenceOfBACnetSpecialEvent" />
        <xs:element name="SequenceOfBACnetTimerState" type="SequenceOfBACnetTimerState" />
        <xs:element name="SequenceOfBACnetTimeValue" type="SequenceOfBACnetTimeValue" />
        <xs:element name="SequenceOfBACnetVMACEntry" type="SequenceOfBACnetVMACEntry" />
        <xs:element name="SequenceOfBACnetVTClass" type="SequenceOfBACnetVTClass" />
        <xs:element name="SequenceOfBACnetVTSession" type="SequenceOfBACnetVTSession" />
        <xs:element name="SequenceOfBitString" type="SequenceOfBitString" />
        <xs:element name="SequenceOfCharacterString" type="SequenceOfCharacterString" />
        <xs:element name="SequenceOfOctetString" type="SequenceOfOctetString" />
        <xs:element name="SequenceOfReadAccessResult" type="SequenceOfReadAccessResult" />
        <xs:element name="SequenceOfReadAccessSpecification"

```

```

type="SequenceOfReadAccessSpecification" />
  <xs:element name="SequenceOfREAL" type="SequenceOfREAL" />
  <xs:element name="SequenceOfUnsigned" type="SequenceOfUnsigned" />
  <xs:element name="SequenceOfUnsigned32" type="SequenceOfUnsigned32" />
  <xs:element name="SequenceOfUnsigned8" type="SequenceOfUnsigned8" />
  <xs:element name="Time" type="Time" />
  <xs:element name="Unsigned" type="Unsigned" />
  <xs:element name="Unsigned16" type="Unsigned16" />
  <xs:element name="Unsigned32" type="Unsigned32" />
  <xs:element name="Unsigned8" type="Unsigned8" />
</xs:choice>
</xs:sequence>
</xs:complexType>
<xs:element name="Value" type="Any" />
</xs:schema>

```

Examples

Example 1:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <Null />
</Value>

```

Example 2:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <Boolean>true</Boolean>
</Value>

```

Example 3:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <Unsigned>1234</Unsigned>
</Value>

```

Example 4:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <INTEGER>-799</INTEGER>
</Value>

```

Example 5:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <REAL>1236.567749</REAL>
</Value>

```

Example 6:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <Double>123459.789012</Double>
</Value>

```

Example 7:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <OctetString>00010203040506070809</OctetString>
</Value>

```

Example 8:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <CharacterString>Abcde</CharacterString>
</Value>

```

Example 9:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BitString>101</BitString>
</Value>
```

Example 10:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <Enumerated>4</Enumerated>
</Value>
```

Example 11:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <Date>2022-12-20</Date>
</Value>
```

Example 12:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <Time>22:39:15.980</Time>
</Value>
```

Example 13:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetObjectIdentifier>device-508</BACnetObjectIdentifier>
</Value>
```

BACnetAccessAuthenticationFactorDisable

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetAccessAuthenticationFactorDisable">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none" />
    <xs:enumeration value="disabled" />
    <xs:enumeration value="disabledLost" />
    <xs:enumeration value="disabledStolen" />
    <xs:enumeration value="disabledDamaged" />
    <xs:enumeration value="disabledDestroyed" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetAccessAuthenticationFactorDisable>none</BACnetAccessAuthenticationFactorDisable>
```

BACnetAccessCredentialDisable

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetAccessCredentialDisable">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none" />
    <xs:enumeration value="disable" />
    <xs:enumeration value="disableManual" />
    <xs:enumeration value="disableLockout" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetAccessCredentialDisable>none</BACnetAccessCredentialDisable>
```

BACnetAccessCredentialDisableReason

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetAccessCredentialDisableReason">
  <xs:restriction base="xs:string">
    <xs:enumeration value="disabled" />
    <xs:enumeration value="disabledNeedsProvisioning" />
    <xs:enumeration value="disabledUnassigned" />
    <xs:enumeration value="disabledNotYetActive" />
    <xs:enumeration value="disabledExpired" />
    <xs:enumeration value="disabledLockout" />
    <xs:enumeration value="disabledMaxDays" />
    <xs:enumeration value="disabledMaxUses" />
    <xs:enumeration value="disabledInactivity" />
    <xs:enumeration value="disabledManual" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetAccessCredentialDisableReason>disabled</BACnetAccessCredentialDisableReason>
```

BACnetAccessEvent

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:simpleType name="BACnetAccessEvent">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none" />
    <xs:enumeration value="granted" />
    <xs:enumeration value="muster" />
    <xs:enumeration value="passbackDetected" />
    <xs:enumeration value="duress" />
    <xs:enumeration value="trace" />
    <xs:enumeration value="lockoutMaxAttempts" />
    <xs:enumeration value="lockoutOther" />
    <xs:enumeration value="lockoutRelinquished" />
    <xs:enumeration value="lockedByHigherPriority" />
    <xs:enumeration value="outOfService" />
    <xs:enumeration value="outOfServiceRelinquished" />
    <xs:enumeration value="accompanimentBy" />
    <xs:enumeration value="authenticationFactorRead" />
    <xs:enumeration value="authorizationDelayed" />
    <xs:enumeration value="verificationRequired" />
    <xs:enumeration value="noEntryAfterGranted" />
    <xs:enumeration value="deniedDenyAll" />
    <xs:enumeration value="deniedUnknownCredential" />
    <xs:enumeration value="deniedAuthenticationUnavailable" />
    <xs:enumeration value="deniedAuthenticationFactorTimeout" />
    <xs:enumeration value="deniedIncorrectAuthenticationFactor" />
    <xs:enumeration value="deniedZoneNoAccessRights" />
    <xs:enumeration value="deniedPointNoAccessRights" />
    <xs:enumeration value="deniedNoAccessRights" />
    <xs:enumeration value="deniedOutOfTimeRange" />
    <xs:enumeration value="deniedThreatLevel" />
    <xs:enumeration value="deniedPassback" />
    <xs:enumeration value="deniedUnexpectedLocationUsage" />
    <xs:enumeration value="deniedMaxAttempts" />
    <xs:enumeration value="deniedLowerOccupancyLimit" />
    <xs:enumeration value="deniedUpperOccupancyLimit" />
    <xs:enumeration value="deniedAuthenticationFactorLost" />
    <xs:enumeration value="deniedAuthenticationFactorStolen" />
    <xs:enumeration value="deniedAuthenticationFactorDamaged" />
    <xs:enumeration value="deniedAuthenticationFactorDestroyed" />
    <xs:enumeration value="deniedAuthenticationFactorDisabled" />
    <xs:enumeration value="deniedAuthenticationFactorError" />
    <xs:enumeration value="deniedCredentialUnassigned" />
    <xs:enumeration value="deniedCredentialNotProvisioned" />
    <xs:enumeration value="deniedCredentialNotYetActive" />
    <xs:enumeration value="deniedCredentialExpired" />
    <xs:enumeration value="deniedCredentialManualDisable" />
    <xs:enumeration value="deniedCredentialLockout" />
    <xs:enumeration value="deniedCredentialMaxDays" />
    <xs:enumeration value="deniedCredentialMaxUses" />
    <xs:enumeration value="deniedCredentialInactivity" />
    <xs:enumeration value="deniedCredentialDisabled" />
    <xs:enumeration value="deniedNoAccompaniment" />
    <xs:enumeration value="deniedIncorrectAccompaniment" />
    <xs:enumeration value="deniedLockout" />
    <xs:enumeration value="deniedVerificationFailed" />
    <xs:enumeration value="deniedVerificationTimeout" />
    <xs:enumeration value="deniedOther" />
  </xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<BACnetAccessEvent>none</BACnetAccessEvent>
```

BACnetAccessRule

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetAccessRule">
  <xs:sequence>
    <xs:element name="TimeRangeSpecifier" type="BACnetAccessRule_TimeRangeSpecifier" />
    <xs:element minOccurs="0" name="TimeRange" type="BACnetDeviceObjectPropertyReference" />
    <xs:element name="LocationSpecifier" type="BACnetAccessRule_LocationSpecifier" />
    <xs:element minOccurs="0" name="Location" type="BACnetDeviceObjectReference" />
    <xs:element name="Enable" type="Boolean" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetAccessRule>
  <TimeRangeSpecifier>specified</TimeRangeSpecifier>
  <TimeRange>
    <ObjectIdentifier>device-502</ObjectIdentifier>
    <PropertyIdentifier>action</PropertyIdentifier>
    <PropertyArrayIndex>1237</PropertyArrayIndex>
    <DeviceIdentifier>device-505</DeviceIdentifier>
  </TimeRange>
  <LocationSpecifier>all</LocationSpecifier>
  <Location>
    <DeviceIdentifier>device-507</DeviceIdentifier>
    <ObjectIdentifier>device-508</ObjectIdentifier>
  </Location>
  <Enable>true</Enable>
</BACnetAccessRule>
```

Example 2:

```
<BACnetAccessRule>
  <TimeRangeSpecifier>specified</TimeRangeSpecifier>
  <LocationSpecifier>all</LocationSpecifier>
  <Enable>true</Enable>
</BACnetAccessRule>
```

BACnetAccessRule_LocationSpecifier

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetAccessRule_LocationSpecifier">
  <xs:restriction base="xs:string">
    <xs:enumeration value="specified" />
    <xs:enumeration value="all" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetAccessRule_LocationSpecifier>specified</BACnetAccessRule_LocationSpecifier>
```

BACnetAccessRule_TimeRangeSpecifier

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetAccessRule_TimeRangeSpecifier">
  <xs:restriction base="xs:string">
    <xs:enumeration value="specified" />
    <xs:enumeration value="always" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetAccessRule_TimeRangeSpecifier>specified</BACnetAccessRule_TimeRangeSpecifier>
```

BACnetAccessZoneOccupancyState

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetAccessZoneOccupancyState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="normal" />
    <xs:enumeration value="belowLowerLimit" />
    <xs:enumeration value="atLowerLimit" />
    <xs:enumeration value="atUpperLimit" />
    <xs:enumeration value="aboveUpperLimit" />
    <xs:enumeration value="disabled" />
    <xs:enumeration value="notSupported" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetAccessZoneOccupancyState>normal</BACnetAccessZoneOccupancyState>
```

BACnetAction

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetAction">
  <xs:restriction base="xs:string">
    <xs:enumeration value="direct" />
    <xs:enumeration value="reverse" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetAction>direct</BACnetAction>
```

BACnetActionCommand

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetActionCommand">
  <xs:sequence>
    <xs:element minOccurs="0" name="DeviceIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="ObjectIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="PropertyIdentifier" type="BACnetPropertyIdentifier" />
    <xs:element minOccurs="0" name="PropertyArrayIndex" type="Unsigned" />
    <xs:element name="PropertyValue" type="Any" />
    <xs:element minOccurs="0" name="Priority" type="Unsigned" />
    <xs:element minOccurs="0" name="PostDelay" type="Unsigned" />
    <xs:element name="QuitOnFailure" type="Boolean" />
    <xs:element name="WriteSuccessful" type="Boolean" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetActionCommand>
  <DeviceIdentifier>device-501</DeviceIdentifier>
  <ObjectIdentifier>device-502</ObjectIdentifier>
  <PropertyIdentifier>action</PropertyIdentifier>
  <PropertyArrayIndex>1237</PropertyArrayIndex>
  <PropertyValue>
    <Boolean>true</Boolean>
  </PropertyValue>
  <Priority>1238</Priority>
  <PostDelay>1239</PostDelay>
  <QuitOnFailure>true</QuitOnFailure>
  <WriteSuccessful>true</WriteSuccessful>
</BACnetActionCommand>
```

Example 2:

```
<BACnetActionCommand>
  <ObjectIdentifier>device-507</ObjectIdentifier>
  <PropertyIdentifier>alarmValues</PropertyIdentifier>
  <PropertyValue>
    <Boolean>true</Boolean>
  </PropertyValue>
  <QuitOnFailure>true</QuitOnFailure>
  <WriteSuccessful>true</WriteSuccessful>
</BACnetActionCommand>
```

BACnetActionList

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetActionList">
  <xs:sequence>
    <xs:element name="Action" type="SequenceOfBACnetActionCommand" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetActionList>
  <Action />
</BACnetActionList>
```

Example 2:

```

<BACnetActionList>
  <Action>
    <BACnetActionCommand>
      <DeviceIdentifier>device-501</DeviceIdentifier>
      <ObjectIdentifier>device-502</ObjectIdentifier>
      <PropertyIdentifier>action</PropertyIdentifier>
      <PropertyArrayIndex>1237</PropertyArrayIndex>
      <PropertyValue>
        <Boolean>true</Boolean>
      </PropertyValue>
      <Priority>1238</Priority>
      <PostDelay>1239</PostDelay>
      <QuitOnFailure>true</QuitOnFailure>
      <WriteSuccessful>true</WriteSuccessful>
    </BACnetActionCommand>
    <BACnetActionCommand>
      <DeviceIdentifier>device-507</DeviceIdentifier>
      <ObjectIdentifier>device-508</ObjectIdentifier>
      <PropertyIdentifier>all</PropertyIdentifier>
      <PropertyArrayIndex>1243</PropertyArrayIndex>
      <PropertyValue>
        <Boolean>true</Boolean>
      </PropertyValue>
      <Priority>1244</Priority>
      <PostDelay>1245</PostDelay>
      <QuitOnFailure>true</QuitOnFailure>
      <WriteSuccessful>true</WriteSuccessful>
    </BACnetActionCommand>
    <BACnetActionCommand>
      <DeviceIdentifier>device-513</DeviceIdentifier>
      <ObjectIdentifier>device-514</ObjectIdentifier>
      <PropertyIdentifier>bias</PropertyIdentifier>
      <PropertyArrayIndex>1249</PropertyArrayIndex>
      <PropertyValue>
        <Boolean>true</Boolean>
      </PropertyValue>
      <Priority>1250</Priority>
      <PostDelay>1251</PostDelay>
      <QuitOnFailure>true</QuitOnFailure>
      <WriteSuccessful>true</WriteSuccessful>
    </BACnetActionCommand>
  </Action>
</BACnetActionList>

```

BACnetAddress

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.valueSource**
- **accessDoor.valueSourceArray**
- **analogOutput.valueSource**
- **analogOutput.valueSourceArray**
- **analogValue.valueSource**
- **analogValue.valueSourceArray**
- **binaryLightingOutput.valueSource**
- **binaryLightingOutput.valueSourceArray**
- **binaryOutput.valueSource**
- **binaryOutput.valueSourceArray**

- `binaryValue.valueSource`
- `binaryValue.valueSourceArray`
- `bitstringValue.valueSource`
- `bitstringValue.valueSourceArray`
- `channel.valueSource`
- `characterstringValue.valueSource`
- `characterstringValue.valueSourceArray`
- `command.valueSource`
- `dateValue.valueSource`
- `dateValue.valueSourceArray`
- `datepatternValue.valueSource`
- `datepatternValue.valueSourceArray`
- `datetimeValue.valueSource`
- `datetimeValue.valueSourceArray`
- `datetimepatternValue.valueSource`
- `datetimepatternValue.valueSourceArray`
- `integerValue.valueSource`
- `integerValue.valueSourceArray`
- `largeAnalogValue.valueSource`
- `largeAnalogValue.valueSourceArray`
- `lifeSafetyPoint.valueSource`
- `lifeSafetyZone.valueSource`
- `lightingOutput.valueSource`
- `lightingOutput.valueSourceArray`
- `loadControl.valueSource`
- `multiStateOutput.valueSource`
- `multiStateOutput.valueSourceArray`
- `multiStateValue.valueSource`
- `multiStateValue.valueSourceArray`
- `octetstringValue.valueSource`
- `octetstringValue.valueSourceArray`
- `positiveIntegerValue.valueSource`

- **positiveIntegerValue.valueSourceArray**
- **timeValue.valueSource**
- **timeValue.valueSourceArray**
- **timepatternValue.valueSource**
- **timepatternValue.valueSourceArray**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetAddress">
    <xs:sequence>
      <xs:element name="NetworkNumber" type="Unsigned16" />
      <xs:element name="MacAddress" type="OctetString" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetAddress" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetAddress>
    <NetworkNumber>7634</NetworkNumber>
    <MacAddress>00010203040506070809</MacAddress>
  </BACnetAddress>
</Value>
```

BACnetAddressBinding

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetAddressBinding">
  <xs:sequence>
    <xs:element name="DeviceIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="DeviceAddress" type="BACnetAddress" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetAddressBinding>
  <DeviceIdentifier>device-501</DeviceIdentifier>
  <DeviceAddress>
    <NetworkNumber>7635</NetworkNumber>
    <MacAddress>00010203040506070809</MacAddress>
  </DeviceAddress>
</BACnetAddressBinding>
```

BACnetAssignedAccessRights

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetAssignedAccessRights">
  <xs:sequence>
    <xs:element name="AssignedAccessRights" type="BACnetDeviceObjectReference" />
    <xs:element name="Enable" type="Boolean" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetAssignedAccessRights>
  <AssignedAccessRights>
    <DeviceIdentifier>device-501</DeviceIdentifier>
    <ObjectIdentifier>device-502</ObjectIdentifier>
  </AssignedAccessRights>
  <Enable>true</Enable>
</BACnetAssignedAccessRights>
```

BACnetAssignedLandingCalls

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetAssignedLandingCalls">
  <xs:sequence>
    <xs:element name="LandingCalls" type="BACnetAssignedLandingCalls_LandingCalls" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetAssignedLandingCalls>
  <LandingCalls />
</BACnetAssignedLandingCalls>
```

Example 2:

```
<BACnetAssignedLandingCalls>
  <LandingCalls>
    <LandingCalls_Item>
      <FloorNumber>33</FloorNumber>
      <Direction>none</Direction>
    </LandingCalls_Item>
    <LandingCalls_Item>
      <FloorNumber>35</FloorNumber>
      <Direction>up</Direction>
    </LandingCalls_Item>
    <LandingCalls_Item>
      <FloorNumber>37</FloorNumber>
      <Direction>upAndDown</Direction>
    </LandingCalls_Item>
  </LandingCalls>
</BACnetAssignedLandingCalls>
```

BACnetAssignedLandingCalls_LandingCalls

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetAssignedLandingCalls_LandingCalls">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="LandingCalls_Item"
type="BACnetAssignedLandingCalls_LandingCalls_Item" />
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```

<SequenceOfBACnetAssignedLandingCalls_LandingCalls />

```

Example 2:

```

<SequenceOfBACnetAssignedLandingCalls_LandingCalls>
  <LandingCalls />
  <LandingCalls>
    <LandingCalls_Item>
      <FloorNumber>33</FloorNumber>
      <Direction>none</Direction>
    </LandingCalls_Item>
  </LandingCalls>
  <LandingCalls>
    <LandingCalls_Item>
      <FloorNumber>35</FloorNumber>
      <Direction>up</Direction>
    </LandingCalls_Item>
  </LandingCalls>
</SequenceOfBACnetAssignedLandingCalls_LandingCalls>

```

Example 3:

```

<SequenceOfBACnetAssignedLandingCalls_LandingCalls>
  <LandingCalls>
    <LandingCalls_Item>
      <FloorNumber>37</FloorNumber>
      <Direction>upAndDown</Direction>
    </LandingCalls_Item>
    <LandingCalls_Item>
      <FloorNumber>39</FloorNumber>
      <Direction>none</Direction>
    </LandingCalls_Item>
    <LandingCalls_Item>
      <FloorNumber>41</FloorNumber>
      <Direction>up</Direction>
    </LandingCalls_Item>
  </LandingCalls>
  <LandingCalls>
    <LandingCalls_Item>
      <FloorNumber>43</FloorNumber>
      <Direction>upAndDown</Direction>
    </LandingCalls_Item>
  </LandingCalls>
  <LandingCalls>
    <LandingCalls_Item>
      <FloorNumber>45</FloorNumber>
      <Direction>none</Direction>
    </LandingCalls_Item>
  </LandingCalls>
</SequenceOfBACnetAssignedLandingCalls_LandingCalls>

```

BACnetAssignedLandingCalls_LandingCalls_Item

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetAssignedLandingCalls_LandingCalls_Item">
  <xs:sequence>
    <xs:element name="FloorNumber" type="Unsigned8" />
    <xs:element name="Direction" type="BACnetLiftCarDirection" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetAssignedLandingCalls_LandingCalls_Item>
  <FloorNumber>33</FloorNumber>
  <Direction>none</Direction>
</BACnetAssignedLandingCalls_LandingCalls_Item>
```

BACnetAuthenticationFactor

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetAuthenticationFactor">
  <xs:sequence>
    <xs:element name="FormatType" type="BACnetAuthenticationFactorType" />
    <xs:element name="FormatClass" type="Unsigned" />
    <xs:element name="Value" type="OctetString" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetAuthenticationFactor>
  <FormatType>undefined</FormatType>
  <FormatClass>1235</FormatClass>
  <Value>00010203040506070809</Value>
</BACnetAuthenticationFactor>
```

BACnetAuthenticationFactorFormat

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetAuthenticationFactorFormat">
  <xs:sequence>
    <xs:element name="FormatType" type="BACnetAuthenticationFactorType" />
    <xs:element minOccurs="0" name="VendorId" type="Unsigned16" />
    <xs:element minOccurs="0" name="VendorFormat" type="Unsigned16" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetAuthenticationFactorFormat>
  <FormatType>undefined</FormatType>
  <VendorId>7635</VendorId>
  <VendorFormat>7636</VendorFormat>
</BACnetAuthenticationFactorFormat>
```

Example 2:

```
<BACnetAuthenticationFactorFormat>
  <FormatType>simpleNumber16</FormatType>
</BACnetAuthenticationFactorFormat>
```

BACnetAuthenticationFactorType

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetAuthenticationFactorType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="undefined" />
    <xs:enumeration value="error" />
    <xs:enumeration value="custom" />
    <xs:enumeration value="simpleNumber16" />
    <xs:enumeration value="simpleNumber32" />
    <xs:enumeration value="simpleNumber56" />
    <xs:enumeration value="simpleAlphaNumeric" />
    <xs:enumeration value="abaTrack2" />
    <xs:enumeration value="wiegand26" />
    <xs:enumeration value="wiegand37" />
    <xs:enumeration value="wiegand37Facility" />
    <xs:enumeration value="facility16Card32" />
    <xs:enumeration value="facility32Card32" />
    <xs:enumeration value="fascN" />
    <xs:enumeration value="fascNBcd" />
    <xs:enumeration value="fascNLarge" />
    <xs:enumeration value="fascNLargeBcd" />
    <xs:enumeration value="gsa75" />
    <xs:enumeration value="chuid" />
    <xs:enumeration value="chuidFull" />
    <xs:enumeration value="guid" />
    <xs:enumeration value="cbeffA" />
    <xs:enumeration value="cbeffB" />
    <xs:enumeration value="cbeffC" />
    <xs:enumeration value="userPassword" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetAuthenticationFactorType>undefined</BACnetAuthenticationFactorType>
```

BACnetAuthenticationPolicy

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetAuthenticationPolicy">
  <xs:sequence>
    <xs:element name="Policy" type="BACnetAuthenticationPolicy_Policy" />
    <xs:element name="OrderEnforced" type="Boolean" />
    <xs:element name="Timeout" type="Unsigned" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetAuthenticationPolicy>
  <Policy />
  <OrderEnforced>true</OrderEnforced>
  <Timeout>1234</Timeout>
</BACnetAuthenticationPolicy>
```

Example 2:


```

<BACnetAuthenticationPolicy>
  <Policy>
    <Policy_Item>
      <CredentialDataInput>
        <DeviceIdentifier>device-502</DeviceIdentifier>
        <ObjectIdentifier>device-503</ObjectIdentifier>
      </CredentialDataInput>
      <Index>1237</Index>
    </Policy_Item>
    <Policy_Item>
      <CredentialDataInput>
        <DeviceIdentifier>device-505</DeviceIdentifier>
        <ObjectIdentifier>device-506</ObjectIdentifier>
      </CredentialDataInput>
      <Index>1240</Index>
    </Policy_Item>
    <Policy_Item>
      <CredentialDataInput>
        <DeviceIdentifier>device-508</DeviceIdentifier>
        <ObjectIdentifier>device-509</ObjectIdentifier>
      </CredentialDataInput>
      <Index>1243</Index>
    </Policy_Item>
  </Policy>
  <OrderEnforced>true</OrderEnforced>
  <Timeout>1244</Timeout>
</BACnetAuthenticationPolicy>

```

BACnetAuthenticationPolicy_Policy

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetAuthenticationPolicy_Policy">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Policy_Item"
type="BACnetAuthenticationPolicy_Policy_Item" />
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```
<SequenceOfBACnetAuthenticationPolicy_Policy />
```

Example 2:

```

<SequenceOfBACnetAuthenticationPolicy_Policy>
  <Policy />
  <Policy>
    <Policy_Item>
      <CredentialDataInput>
        <DeviceIdentifier>device-501</DeviceIdentifier>
        <ObjectIdentifier>device-502</ObjectIdentifier>
      </CredentialDataInput>
      <Index>1236</Index>
    </Policy_Item>
  </Policy>
  <Policy>
    <Policy_Item>
      <CredentialDataInput>
        <DeviceIdentifier>device-504</DeviceIdentifier>
        <ObjectIdentifier>device-505</ObjectIdentifier>
      </CredentialDataInput>
      <Index>1239</Index>
    </Policy_Item>
  </Policy>
</SequenceOfBACnetAuthenticationPolicy_Policy>

```

Example 3:

```

<SequenceOfBACnetAuthenticationPolicy_Policy>
  <Policy>
    <Policy_Item>
      <CredentialDataInput>
        <DeviceIdentifier>device-507</DeviceIdentifier>
        <ObjectIdentifier>device-508</ObjectIdentifier>
      </CredentialDataInput>
      <Index>1242</Index>
    </Policy_Item>
    <Policy_Item>
      <CredentialDataInput>
        <DeviceIdentifier>device-510</DeviceIdentifier>
        <ObjectIdentifier>device-511</ObjectIdentifier>
      </CredentialDataInput>
      <Index>1245</Index>
    </Policy_Item>
    <Policy_Item>
      <CredentialDataInput>
        <DeviceIdentifier>device-513</DeviceIdentifier>
        <ObjectIdentifier>device-514</ObjectIdentifier>
      </CredentialDataInput>
      <Index>1248</Index>
    </Policy_Item>
  </Policy>
  <Policy>
    <Policy_Item>
      <CredentialDataInput>
        <DeviceIdentifier>device-516</DeviceIdentifier>
        <ObjectIdentifier>device-517</ObjectIdentifier>
      </CredentialDataInput>
      <Index>1251</Index>
    </Policy_Item>
  </Policy>
  <Policy>
    <Policy_Item>
      <CredentialDataInput>
        <DeviceIdentifier>device-519</DeviceIdentifier>
        <ObjectIdentifier>device-520</ObjectIdentifier>
      </CredentialDataInput>
      <Index>1254</Index>
    </Policy_Item>
  </Policy>
</SequenceOfBACnetAuthenticationPolicy_Policy>

```

BACnetAuthenticationPolicy_Policy_Item

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetAuthenticationPolicy_Policy_Item">
  <xs:sequence>
    <xs:element name="CredentialDataInput" type="BACnetDeviceObjectReference" />
    <xs:element name="Index" type="Unsigned" />
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```

<BACnetAuthenticationPolicy_Policy_Item>
  <CredentialDataInput>
    <DeviceIdentifier>device-501</DeviceIdentifier>
    <ObjectIdentifier>device-502</ObjectIdentifier>
  </CredentialDataInput>
  <Index>1236</Index>
</BACnetAuthenticationPolicy_Policy_Item>

```

BACnetAuthenticationStatus

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetAuthenticationStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="notReady" />
    <xs:enumeration value="ready" />
    <xs:enumeration value="disabled" />
    <xs:enumeration value="waitingForAuthenticationFactor" />
    <xs:enumeration value="waitingForAccompaniment" />
    <xs:enumeration value="waitingForVerification" />
    <xs:enumeration value="inProgress" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetAuthenticationStatus>notReady</BACnetAuthenticationStatus>
```

BACnetAuthorizationExemption

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetAuthorizationExemption">
  <xs:restriction base="xs:string">
    <xs:enumeration value="passback" />
    <xs:enumeration value="occupancyCheck" />
    <xs:enumeration value="accessRights" />
    <xs:enumeration value="lockout" />
    <xs:enumeration value="deny" />
    <xs:enumeration value="verification" />
    <xs:enumeration value="authorizationDelay" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetAuthorizationExemption>passback</BACnetAuthorizationExemption>
```

BACnetBDTEntry

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetBDTEntry">
  <xs:sequence>
    <xs:element name="BbmdAddress" type="BACnetHostNPort" />
    <xs:element minOccurs="0" name="BroadcastMask" type="OctetString" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetBDTEntry>
  <BbmdAddress>
    <Host>
      <None />
    </Host>
    <Port>7634</Port>
  </BbmdAddress>
  <BroadcastMask>00010203040506070809</BroadcastMask>
</BACnetBDTEntry>
```

Example 2:

```
<BACnetBDTEntry>
  <BbmdAddress>
    <Host>
      <None />
    </Host>
    <Port>7635</Port>
  </BbmdAddress>
</BACnetBDTEntry>
```

BACnetBackupState

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetBackupState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="idle" />
    <xs:enumeration value="preparingForBackup" />
    <xs:enumeration value="preparingForRestore" />
    <xs:enumeration value="performingABackup" />
    <xs:enumeration value="performingARestore" />
    <xs:enumeration value="backupFailure" />
    <xs:enumeration value="restoreFailure" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetBackupState>idle</BACnetBackupState>
```

BACnetBinaryLightingPV

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetBinaryLightingPV">
  <xs:restriction base="xs:string">
    <xs:enumeration value="off" />
    <xs:enumeration value="on" />
    <xs:enumeration value="warn" />
    <xs:enumeration value="warnOff" />
    <xs:enumeration value="warnRelinquish" />
    <xs:enumeration value="stop" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetBinaryLightingPV>off</BACnetBinaryLightingPV>
```

BACnetBinaryPV

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetBinaryPV">
  <xs:restriction base="xs:string">
    <xs:enumeration value="inactive" />
    <xs:enumeration value="active" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetBinaryPV>inactive</BACnetBinaryPV>
```

BACnetCOVMultipleSubscription

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetCOVMultipleSubscription">
  <xs:sequence>
    <xs:element name="Recipient" type="BACnetRecipientProcess" />
    <xs:element name="IssueConfirmedNotifications" type="Boolean" />
    <xs:element name="TimeRemaining" type="Unsigned" />
    <xs:element name="MaxNotificationDelay" type="Unsigned" />
    <xs:element name="ListOfCovSubscriptionSpecifications"
type="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetCOVMultipleSubscription>
  <Recipient>
    <Recipient>
      <Device>device-501</Device>
    </Recipient>
    <ProcessIdentifier>2346</ProcessIdentifier>
  </Recipient>
  <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
  <TimeRemaining>1236</TimeRemaining>
  <MaxNotificationDelay>1237</MaxNotificationDelay>
  <ListOfCovSubscriptionSpecifications />
</BACnetCOVMultipleSubscription>
```

Example 2:

```

<BACnetCOVMultipleSubscription>
  <Recipient>
    <Recipient>
      <Device>device-505</Device>
    </Recipient>
    <ProcessIdentifier>2350</ProcessIdentifier>
  </Recipient>
  <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
  <TimeRemaining>1240</TimeRemaining>
  <MaxNotificationDelay>1241</MaxNotificationDelay>
  <ListOfCovSubscriptionSpecifications>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-509</MonitoredObjectIdentifier>
      <ListOfCovReferences />
    </ListOfCovSubscriptionSpecifications_Item>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-510</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>apduSegmentTimeout</PropertyIdentifier>
            <PropertyArrayIndex>1245</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1246.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-514</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>bias</PropertyIdentifier>
            <PropertyArrayIndex>1249</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1250.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>
</BACnetCOVMultipleSubscription>

```

Example 3:

```

<BACnetCOVMultipleSubscription>
  <Recipient>
    <Recipient>
      <Device>device-518</Device>
    </Recipient>
    <ProcessIdentifier>2363</ProcessIdentifier>
  </Recipient>
  <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
  <TimeRemaining>1253</TimeRemaining>
  <MaxNotificationDelay>1254</MaxNotificationDelay>
  <ListOfCovSubscriptionSpecifications>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-522</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>dateList</PropertyIdentifier>
            <PropertyArrayIndex>1257</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1258.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>derivativeConstant</PropertyIdentifier>
            <PropertyArrayIndex>1260</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1261.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>descriptionOfHalt</PropertyIdentifier>
            <PropertyArrayIndex>1263</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1264.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-532</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>elapsedActiveTime</PropertyIdentifier>
            <PropertyArrayIndex>1267</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1268.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-536</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>eventType</PropertyIdentifier>
            <PropertyArrayIndex>1271</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1272.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>
</BACnetCOVMultipleSubscription>

```

BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="ListOfCovSubscriptionSpecifications_Item"
type="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<SequenceOfBACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications />
```

Example 2:

```
<SequenceOfBACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications>
  <ListOfCovSubscriptionSpecifications />
  <ListOfCovSubscriptionSpecifications>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-501</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>ackRequired</PropertyIdentifier>
            <PropertyArrayIndex>1236</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1237.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>
  <ListOfCovSubscriptionSpecifications>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-505</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>activeVtSessions</PropertyIdentifier>
            <PropertyArrayIndex>1240</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1241.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>
</SequenceOfBACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications>
```

Example 3:


```

<SequenceOfBACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications>
  <ListOfCovSubscriptionSpecifications>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-509</MonitoredObjectIdentifier>
      <ListOfCovReferences />
    </ListOfCovSubscriptionSpecifications_Item>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-510</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>apduSegmentTimeout</PropertyIdentifier>
            <PropertyArrayIndex>1245</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1246.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-514</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>bias</PropertyIdentifier>
            <PropertyArrayIndex>1249</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1250.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>
  <ListOfCovSubscriptionSpecifications>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-518</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>controlledVariableReference</PropertyIdentifier>
            <PropertyArrayIndex>1253</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1254.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>
  <ListOfCovSubscriptionSpecifications>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-522</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>dateList</PropertyIdentifier>
            <PropertyArrayIndex>1257</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1258.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>
</SequenceOfBACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications>

```

Example 4:

```

<SequenceOfBACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications>
  <ListOfCovSubscriptionSpecifications>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-526</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>derivativeConstantUnits</PropertyIdentifier>
            <PropertyArrayIndex>1261</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1262.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>deviceAddressBinding</PropertyIdentifier>
            <PropertyArrayIndex>1264</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1265.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>elapsedActiveTime</PropertyIdentifier>
            <PropertyArrayIndex>1267</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1268.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-536</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>eventType</PropertyIdentifier>
            <PropertyArrayIndex>1271</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1272.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-540</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>fileAccessMethod</PropertyIdentifier>
            <PropertyArrayIndex>1275</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1276.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>
  <ListOfCovSubscriptionSpecifications>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-544</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>highLimit</PropertyIdentifier>
            <PropertyArrayIndex>1279</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1280.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>

```

```

    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>
  <ListOfCovSubscriptionSpecifications>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-548</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>integralConstant</PropertyIdentifier>
            <PropertyArrayIndex>1283</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1284.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>
</SequenceOfBACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications>

```

BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item">
  <xs:sequence>
    <xs:element name="MonitoredObjectIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="ListOfCovReferences"
type="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences" />
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```

<BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item>
  <MonitoredObjectIdentifier>device-501</MonitoredObjectIdentifier>
  <ListOfCovReferences />
</BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item>

```

Example 2:

```
<BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item>
  <MonitoredObjectIdentifier>device-502</MonitoredObjectIdentifier>
  <ListOfCovReferences>
    <ListOfCovReferences_Item>
      <MonitoredProperty>
        <PropertyIdentifier>action</PropertyIdentifier>
        <PropertyArrayIndex>1237</PropertyArrayIndex>
      </MonitoredProperty>
      <CovIncrement>1238.567749</CovIncrement>
      <Timestamped>true</Timestamped>
    </ListOfCovReferences_Item>
    <ListOfCovReferences_Item>
      <MonitoredProperty>
        <PropertyIdentifier>activeVtSessions</PropertyIdentifier>
        <PropertyArrayIndex>1240</PropertyArrayIndex>
      </MonitoredProperty>
      <CovIncrement>1241.567749</CovIncrement>
      <Timestamped>true</Timestamped>
    </ListOfCovReferences_Item>
    <ListOfCovReferences_Item>
      <MonitoredProperty>
        <PropertyIdentifier>all</PropertyIdentifier>
        <PropertyArrayIndex>1243</PropertyArrayIndex>
      </MonitoredProperty>
      <CovIncrement>1244.567749</CovIncrement>
      <Timestamped>true</Timestamped>
    </ListOfCovReferences_Item>
  </ListOfCovReferences>
</BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item>
```

BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType
  name="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="ListOfCovReferences_Item"
      type="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences_Item" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<SequenceOfBACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences />
```

Example 2:

```

<SequenceOfBACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences>
<ListOfCovReferences />
<ListOfCovReferences>
  <ListOfCovReferences_Item>
    <MonitoredProperty>
      <PropertyIdentifier>ackedTransitions</PropertyIdentifier>
      <PropertyArrayIndex>1235</PropertyArrayIndex>
    </MonitoredProperty>
    <CovIncrement>1236.567749</CovIncrement>
    <Timestamped>true</Timestamped>
  </ListOfCovReferences_Item>
</ListOfCovReferences>
<ListOfCovReferences>
  <ListOfCovReferences_Item>
    <MonitoredProperty>
      <PropertyIdentifier>actionText</PropertyIdentifier>
      <PropertyArrayIndex>1238</PropertyArrayIndex>
    </MonitoredProperty>
    <CovIncrement>1239.567749</CovIncrement>
    <Timestamped>true</Timestamped>
  </ListOfCovReferences_Item>
</ListOfCovReferences>
</SequenceOfBACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences>

```

Example 3:

```
<SequenceOfBACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReference
S>
  <ListOfCovReferences>
    <ListOfCovReferences_Item>
      <MonitoredProperty>
        <PropertyIdentifier>alarmValue</PropertyIdentifier>
        <PropertyArrayIndex>1241</PropertyArrayIndex>
      </MonitoredProperty>
      <CovIncrement>1242.567749</CovIncrement>
      <Timestamped>true</Timestamped>
    </ListOfCovReferences_Item>
    <ListOfCovReferences_Item>
      <MonitoredProperty>
        <PropertyIdentifier>allWritesSuccessful</PropertyIdentifier>
        <PropertyArrayIndex>1244</PropertyArrayIndex>
      </MonitoredProperty>
      <CovIncrement>1245.567749</CovIncrement>
      <Timestamped>true</Timestamped>
    </ListOfCovReferences_Item>
    <ListOfCovReferences_Item>
      <MonitoredProperty>
        <PropertyIdentifier>applicationSoftwareVersion</PropertyIdentifier>
        <PropertyArrayIndex>1247</PropertyArrayIndex>
      </MonitoredProperty>
      <CovIncrement>1248.567749</CovIncrement>
      <Timestamped>true</Timestamped>
    </ListOfCovReferences_Item>
  </ListOfCovReferences>
  <ListOfCovReferences>
    <ListOfCovReferences_Item>
      <MonitoredProperty>
        <PropertyIdentifier>changeOfStateCount</PropertyIdentifier>
        <PropertyArrayIndex>1250</PropertyArrayIndex>
      </MonitoredProperty>
      <CovIncrement>1251.567749</CovIncrement>
      <Timestamped>true</Timestamped>
    </ListOfCovReferences_Item>
  </ListOfCovReferences>
  <ListOfCovReferences>
    <ListOfCovReferences_Item>
      <MonitoredProperty>
        <PropertyIdentifier>controlledVariableReference</PropertyIdentifier>
        <PropertyArrayIndex>1253</PropertyArrayIndex>
      </MonitoredProperty>
      <CovIncrement>1254.567749</CovIncrement>
      <Timestamped>true</Timestamped>
    </ListOfCovReferences_Item>
  </ListOfCovReferences>
</SequenceOfBACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferenc
es>
```

BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences_Item

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType
name="BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences_Ite
m">
  <xs:sequence>
    <xs:element name="MonitoredProperty" type="BACnetPropertyReference" />
    <xs:element minOccurs="0" name="CovIncrement" type="REAL" />
    <xs:element name="Timestamped" type="Boolean" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences_Item>
  <MonitoredProperty>
    <PropertyIdentifier>ackedTransitions</PropertyIdentifier>
    <PropertyArrayIndex>1235</PropertyArrayIndex>
  </MonitoredProperty>
  <CovIncrement>1236.567749</CovIncrement>
  <Timestamped>true</Timestamped>
</BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences_Item>
```

Example 2:

```
<BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences_Item>
  <MonitoredProperty>
    <PropertyIdentifier>actionText</PropertyIdentifier>
    <PropertyArrayIndex>1238</PropertyArrayIndex>
  </MonitoredProperty>
  <Timestamped>true</Timestamped>
</BACnetCOVMultipleSubscription_ListOfCovSubscriptionSpecifications_Item_ListOfCovReferences_Item>
```

BACnetCOVSubscription

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetCOVSubscription">
  <xs:sequence>
    <xs:element name="Recipient" type="BACnetRecipientProcess" />
    <xs:element name="MonitoredPropertyReference" type="BACnetObjectPropertyReference" />
    <xs:element name="IssueConfirmedNotifications" type="Boolean" />
    <xs:element name="TimeRemaining" type="Unsigned" />
    <xs:element minOccurs="0" name="CovIncrement" type="REAL" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetCOVSubscription>
  <Recipient>
    <Recipient>
      <Device>device-501</Device>
    </Recipient>
    <ProcessIdentifier>2346</ProcessIdentifier>
  </Recipient>
  <MonitoredPropertyReference>
    <ObjectIdentifier>device-503</ObjectIdentifier>
    <PropertyIdentifier>actionText</PropertyIdentifier>
    <PropertyArrayIndex>1238</PropertyArrayIndex>
  </MonitoredPropertyReference>
  <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
  <TimeRemaining>1239</TimeRemaining>
  <CovIncrement>1240.567749</CovIncrement>
</BACnetCOVSubscription>
```

Example 2:

```
<BACnetCOVSubscription>
  <Recipient>
    <Recipient>
      <Device>device-508</Device>
    </Recipient>
    <ProcessIdentifier>2353</ProcessIdentifier>
  </Recipient>
  <MonitoredPropertyReference>
    <ObjectIdentifier>device-510</ObjectIdentifier>
    <PropertyIdentifier>apduSegmentTimeout</PropertyIdentifier>
    <PropertyArrayIndex>1245</PropertyArrayIndex>
  </MonitoredPropertyReference>
  <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
  <TimeRemaining>1246</TimeRemaining>
</BACnetCOVSubscription>
```

BACnetCalendarEntry

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetCalendarEntry">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Date" type="Date" />
      <xs:element name="DateRange" type="BACnetDateRange" />
      <xs:element name="WeekNDay" type="BACnetWeekNDay" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetCalendarEntry>
  <Date>2017-12-20</Date>
</BACnetCalendarEntry>
```

Example 2:

```
<BACnetCalendarEntry>
  <DateRange>
    <StartDate>2018-12-20</StartDate>
    <EndDate>2019-12-20</EndDate>
  </DateRange>
</BACnetCalendarEntry>
```

Example 3:

```
<BACnetCalendarEntry>
  <WeekNDay>FFFFFF</WeekNDay>
</BACnetCalendarEntry>
```

BACnetCredentialAuthenticationFactor

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetCredentialAuthenticationFactor">
  <xs:sequence>
    <xs:element name="Disable" type="BACnetAccessAuthenticationFactorDisable" />
    <xs:element name="AuthenticationFactor" type="BACnetAuthenticationFactor" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:


```

<BACnetCredentialAuthenticationFactor>
  <Disable>none</Disable>
  <AuthenticationFactor>
    <FormatType>error</FormatType>
    <FormatClass>1236</FormatClass>
    <Value>00010203040506070809</Value>
  </AuthenticationFactor>
</BACnetCredentialAuthenticationFactor>

```

BACnetDailySchedule

This XML representation of complex data is used by the following objects/properties:

- **schedule.weeklySchedule**

XML Schema

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetDailySchedule">
    <xs:sequence>
      <xs:element name="DaySchedule" type="SequenceOfBACnetTimeValue" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetDailySchedule" />
</xs:schema>

```

Examples

Example 1:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetDailySchedule>
    <DaySchedule />
  </BACnetDailySchedule>
</Value>

```

Example 2:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetDailySchedule>
    <DaySchedule>
      <BACnetTimeValue>
        <Time>22:39:15.980</Time>
        <Value>
          <Boolean>true</Boolean>
        </Value>
      </BACnetTimeValue>
      <BACnetTimeValue>
        <Time>22:39:15.980</Time>
        <Value>
          <Boolean>true</Boolean>
        </Value>
      </BACnetTimeValue>
      <BACnetTimeValue>
        <Time>22:39:15.980</Time>
        <Value>
          <Boolean>true</Boolean>
        </Value>
      </BACnetTimeValue>
    </DaySchedule>
  </BACnetDailySchedule>
</Value>

```

BACnetDateRange

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetDateRange">
  <xs:sequence>
    <xs:element name="StartDate" type="Date" />
    <xs:element name="EndDate" type="Date" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetDateRange>
  <StartDate>2017-12-20</StartDate>
  <EndDate>2018-12-20</EndDate>
</BACnetDateRange>
```

BACnetDateTime

This XML representation of complex data is used by the following objects/properties:

- **accessCredential.activationTime**
- **accessCredential.expirationTime**
- **accessCredential.lastUseTime**
- **accessDoor.priorityArray**
- **accessZone.lastCredentialAddedTime**
- **accessZone.lastCredentialRemovedTime**
- **accumulator.loggingRecord**
- **accumulator.valueChangeTime**
- **analogOutput.priorityArray**
- **analogValue.priorityArray**
- **averaging.maximumValueTimestamp**
- **averaging.minimumValueTimestamp**
- **binaryInput.changeOfStateTime**
- **binaryInput.timeOfActiveTimeReset**
- **binaryInput.timeOfStateCountReset**
- **binaryLightingOutput.priorityArray**
- **binaryLightingOutput.timeOfActiveTimeReset**
- **binaryLightingOutput.timeOfStrikeCountReset**
- **binaryOutput.changeOfStateTime**
- **binaryOutput.priorityArray**

- **binaryOutput.timeOfActiveTimeReset**
- **binaryOutput.timeOfStateCountReset**
- **binaryValue.changeOfStateTime**
- **binaryValue.priorityArray**
- **binaryValue.timeOfActiveTimeReset**
- **binaryValue.timeOfStateCountReset**
- **bitstringValue.priorityArray**
- **characterstringValue.priorityArray**
- **dateValue.priorityArray**
- **datepatternValue.priorityArray**
- **datetimeValue.presentValue**
- **datetimeValue.priorityArray**
- **datetimeValue.relinquishDefault**
- **datetimepatternValue.presentValue**
- **datetimepatternValue.priorityArray**
- **datetimepatternValue.relinquishDefault**
- **eventLog.startTime**
- **eventLog.stopTime**
- **file.modificationDate**
- **integerValue.priorityArray**
- **largeAnalogValue.priorityArray**
- **lightingOutput.priorityArray**
- **loadControl.startTime**
- **multiStateOutput.priorityArray**
- **multiStateValue.priorityArray**
- **octetstringValue.priorityArray**
- **positiveIntegerValue.priorityArray**
- **pulseConverter.countChangeTime**
- **pulseConverter.updateTime**
- **timeValue.priorityArray**
- **timepatternValue.priorityArray**
- **timer.expirationTime**

- **timer.stateChangeValues**
- **timer.updateTime**
- **trendLog.startTime**
- **trendLog.stopTime**
- **trendLogMultiple.startTime**
- **trendLogMultiple.stopTime**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetDateTime">
    <xs:sequence>
      <xs:element name="Date" type="Date" />
      <xs:element name="Time" type="Time" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetDateTime" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetDateTime>
    <Date>2017-12-20</Date>
    <Time>22:39:15.980</Time>
  </BACnetDateTime>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetDateTime>
    <Date>????-??-??-?</Date>
    <Time>?:?:?:?.??</Time>
  </BACnetDateTime>
</Value>
```

Example 3:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetDateTime>
    <Date>????-??-??-?</Date>
    <Time>22:39:15.980</Time>
  </BACnetDateTime>
</Value>
```

BACnetDaysOfWeek

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetDaysOfWeek">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="monday" type="Boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="tuesday" type="Boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="wednesday" type="Boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="thursday" type="Boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="friday" type="Boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="saturday" type="Boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="sunday" type="Boolean" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetDaysOfWeek>
  <monday>true</monday>
  <wednesday>true</wednesday>
  <friday>true</friday>
  <sunday>true</sunday>
</BACnetDaysOfWeek>
```

BACnetDestination

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetDestination">
  <xs:sequence>
    <xs:element name="ValidDays" type="BACnetDaysOfWeek" />
    <xs:element name="FromTime" type="Time" />
    <xs:element name="ToTime" type="Time" />
    <xs:element name="Recipient" type="BACnetRecipient" />
    <xs:element name="ProcessIdentifier" type="Unsigned32" />
    <xs:element name="IssueConfirmedNotifications" type="Boolean" />
    <xs:element name="Transitions" type="BACnetEventTransitionBits" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetDestination>
  <ValidDays>
    <monday>true</monday>
    <wednesday>true</wednesday>
    <friday>true</friday>
    <sunday>true</sunday>
  </ValidDays>
  <FromTime>22:39:15.980</FromTime>
  <ToTime>22:39:15.980</ToTime>
  <Recipient>
    <Device>device-503</Device>
  </Recipient>
  <ProcessIdentifier>2348</ProcessIdentifier>
  <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
  <Transitions>
    <toOffnormal>true</toOffnormal>
    <toNormal>true</toNormal>
  </Transitions>
</BACnetDestination>
```

BACnetDeviceObjectPropertyReference

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetDeviceObjectPropertyReference">
  <xs:sequence>
    <xs:element name="ObjectIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="PropertyIdentifier" type="BACnetPropertyIdentifier" />
    <xs:element minOccurs="0" name="PropertyArrayIndex" type="Unsigned" />
    <xs:element minOccurs="0" name="DeviceIdentifier" type="BACnetObjectIdentifier" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetDeviceObjectPropertyReference>
  <ObjectIdentifier>device-501</ObjectIdentifier>
  <PropertyIdentifier>ackRequired</PropertyIdentifier>
  <PropertyArrayIndex>1236</PropertyArrayIndex>
  <DeviceIdentifier>device-504</DeviceIdentifier>
</BACnetDeviceObjectPropertyReference>
```

Example 2:

```
<BACnetDeviceObjectPropertyReference>
  <ObjectIdentifier>device-505</ObjectIdentifier>
  <PropertyIdentifier>activeVtSessions</PropertyIdentifier>
</BACnetDeviceObjectPropertyReference>
```

BACnetDeviceObjectPropertyValue

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetDeviceObjectPropertyValue">
  <xs:sequence>
    <xs:element name="DeviceIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="ObjectIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="PropertyIdentifier" type="BACnetPropertyIdentifier" />
    <xs:element minOccurs="0" name="PropertyArrayIndex" type="Unsigned" />
    <xs:element name="PropertyValue" type="Any" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetDeviceObjectPropertyValue>
  <DeviceIdentifier>device-501</DeviceIdentifier>
  <ObjectIdentifier>device-502</ObjectIdentifier>
  <PropertyIdentifier>action</PropertyIdentifier>
  <PropertyArrayIndex>1237</PropertyArrayIndex>
  <PropertyValue>
    <Boolean>true</Boolean>
  </PropertyValue>
</BACnetDeviceObjectPropertyValue>
```

Example 2:

```
<BACnetDeviceObjectPropertyValue>
  <DeviceIdentifier>device-505</DeviceIdentifier>
  <ObjectIdentifier>device-506</ObjectIdentifier>
  <PropertyIdentifier>alarmValue</PropertyIdentifier>
  <PropertyValue>
    <Boolean>true</Boolean>
  </PropertyValue>
</BACnetDeviceObjectPropertyValue>
```

BACnetDeviceObjectReference

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.valueSource**
- **accessDoor.valueSourceArray**
- **analogOutput.valueSource**
- **analogOutput.valueSourceArray**
- **analogValue.valueSource**
- **analogValue.valueSourceArray**
- **binaryLightingOutput.valueSource**
- **binaryLightingOutput.valueSourceArray**
- **binaryOutput.valueSource**
- **binaryOutput.valueSourceArray**
- **binaryValue.valueSource**
- **binaryValue.valueSourceArray**
- **bitstringValue.valueSource**
- **bitstringValue.valueSourceArray**
- **channel.valueSource**
- **characterstringValue.valueSource**
- **characterstringValue.valueSourceArray**
- **command.valueSource**
- **dateValue.valueSource**
- **dateValue.valueSourceArray**
- **datepatternValue.valueSource**
- **datepatternValue.valueSourceArray**
- **datetimeValue.valueSource**
- **datetimeValue.valueSourceArray**
- **datetimepatternValue.valueSource**
- **datetimepatternValue.valueSourceArray**
- **integerValue.valueSource**
- **integerValue.valueSourceArray**
- **largeAnalogValue.valueSource**
- **largeAnalogValue.valueSourceArray**
- **lifeSafetyPoint.valueSource**
- **lifeSafetyZone.valueSource**

- **lightingOutput.valueSource**
- **lightingOutput.valueSourceArray**
- **loadControl.valueSource**
- **multiStateOutput.valueSource**
- **multiStateOutput.valueSourceArray**
- **multiStateValue.valueSource**
- **multiStateValue.valueSourceArray**
- **octetstringValue.valueSource**
- **octetstringValue.valueSourceArray**
- **positiveIntegerValue.valueSource**
- **positiveIntegerValue.valueSourceArray**
- **timeValue.valueSource**
- **timeValue.valueSourceArray**
- **timepatternValue.valueSource**
- **timepatternValue.valueSourceArray**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetDeviceObjectReference">
    <xs:sequence>
      <xs:element minOccurs="0" name="DeviceIdentifier" type="BACnetObjectIdentifier" />
      <xs:element name="ObjectIdentifier" type="BACnetObjectIdentifier" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetDeviceObjectReference" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetDeviceObjectReference>
    <DeviceIdentifier>device-501</DeviceIdentifier>
    <ObjectIdentifier>device-502</ObjectIdentifier>
  </BACnetDeviceObjectReference>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetDeviceObjectReference>
    <ObjectIdentifier>device-503</ObjectIdentifier>
  </BACnetDeviceObjectReference>
</Value>
```


BACnetDeviceStatus

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetDeviceStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="operational" />
    <xs:enumeration value="operationalReadOnly" />
    <xs:enumeration value="downloadRequired" />
    <xs:enumeration value="downloadInProgress" />
    <xs:enumeration value="nonOperational" />
    <xs:enumeration value="backupInProgress" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetDeviceStatus>operational</BACnetDeviceStatus>
```

BACnetDoorAlarmState

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetDoorAlarmState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="normal" />
    <xs:enumeration value="alarm" />
    <xs:enumeration value="doorOpenTooLong" />
    <xs:enumeration value="forcedOpen" />
    <xs:enumeration value="tamper" />
    <xs:enumeration value="doorFault" />
    <xs:enumeration value="lockDown" />
    <xs:enumeration value="freeAccess" />
    <xs:enumeration value="egressOpen" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetDoorAlarmState>normal</BACnetDoorAlarmState>
```

BACnetDoorSecuredStatus

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetDoorSecuredStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="secured" />
    <xs:enumeration value="unsecured" />
    <xs:enumeration value="unknown" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetDoorSecuredStatus>secured</BACnetDoorSecuredStatus>
```

BACnetDoorStatus

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetDoorStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="closed" />
    <xs:enumeration value="opened" />
    <xs:enumeration value="unknown" />
    <xs:enumeration value="doorFault" />
    <xs:enumeration value="unused" />
    <xs:enumeration value="none" />
    <xs:enumeration value="closing" />
    <xs:enumeration value="opening" />
    <xs:enumeration value="safetyLocked" />
    <xs:enumeration value="limitedOpened" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetDoorStatus>closed</BACnetDoorStatus>
```

BACnetDoorValue

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetDoorValue">
  <xs:restriction base="xs:string">
    <xs:enumeration value="lock" />
    <xs:enumeration value="unlock" />
    <xs:enumeration value="pulseUnlock" />
    <xs:enumeration value="extendedPulseUnlock" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetDoorValue>lock</BACnetDoorValue>
```

BACnetEngineeringUnits

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetEngineeringUnits">
  <xs:restriction base="xs:string">
    <xs:enumeration value="squareMeters" />
    <xs:enumeration value="squareFeet" />
    <xs:enumeration value="milliamperes" />
    <xs:enumeration value="amperes" />
    <xs:enumeration value="ohms" />
    <xs:enumeration value="volts" />
    <xs:enumeration value="kilovolts" />
    <xs:enumeration value="megavolts" />
    <xs:enumeration value="voltAmperes" />
    <xs:enumeration value="kilovoltAmperes" />
    <xs:enumeration value="megavoltAmperes" />
    <xs:enumeration value="voltAmperesReactive" />
    <xs:enumeration value="kilovoltAmperesReactive" />
    <xs:enumeration value="megavoltAmperesReactive" />
    <xs:enumeration value="degreesPhase" />
    <xs:enumeration value="powerFactor" />
    <xs:enumeration value="joules" />
    <xs:enumeration value="kilojoules" />
    <xs:enumeration value="wattHours" />
    <xs:enumeration value="kilowattHours" />
    <xs:enumeration value="btus" />
    <xs:enumeration value="therms" />
    <xs:enumeration value="tonHours" />
    <xs:enumeration value="joulesPerKilogramDryAir" />
    <xs:enumeration value="btusPerPoundDryAir" />
    <xs:enumeration value="cyclesPerHour" />
    <xs:enumeration value="cyclesPerMinute" />
    <xs:enumeration value="hertz" />
    <xs:enumeration value="gramsOfWaterPerKilogramDryAir" />
    <xs:enumeration value="percentRelativeHumidity" />
    <xs:enumeration value="millimeters" />
    <xs:enumeration value="meters" />
    <xs:enumeration value="inches" />
    <xs:enumeration value="feet" />
    <xs:enumeration value="wattsPerSquareFoot" />
    <xs:enumeration value="wattsPerSquareMeter" />
    <xs:enumeration value="lumens" />
    <xs:enumeration value="luxes" />
    <xs:enumeration value="footCandles" />
    <xs:enumeration value="kilograms" />
    <xs:enumeration value="poundsMass" />
    <xs:enumeration value="tons" />
    <xs:enumeration value="kilogramsPerSecond" />
    <xs:enumeration value="kilogramsPerMinute" />
    <xs:enumeration value="kilogramsPerHour" />
    <xs:enumeration value="poundsMassPerMinute" />
    <xs:enumeration value="poundsMassPerHour" />
    <xs:enumeration value="watts" />
    <xs:enumeration value="kilowatts" />
    <xs:enumeration value="megawatts" />
    <xs:enumeration value="btusPerHour" />
    <xs:enumeration value="horsepower" />
    <xs:enumeration value="tonsRefrigeration" />
    <xs:enumeration value="pascals" />
    <xs:enumeration value="kilopascals" />
    <xs:enumeration value="bars" />
    <xs:enumeration value="poundsForcePerSquareInch" />
    <xs:enumeration value="centimetersOfWater" />
    <xs:enumeration value="inchesOfWater" />
    <xs:enumeration value="millimetersOfMercury" />
    <xs:enumeration value="centimetersOfMercury" />
    <xs:enumeration value="inchesOfMercury" />
    <xs:enumeration value="degreesCelsius" />
    <xs:enumeration value="degreesKelvin" />
    <xs:enumeration value="degreesFahrenheit" />
    <xs:enumeration value="degreeDaysCelsius" />
    <xs:enumeration value="degreeDaysFahrenheit" />
    <xs:enumeration value="years" />
    <xs:enumeration value="months" />
  </xs:restriction>
</xs:simpleType>
```

```
<xs:enumeration value="weeks" />
<xs:enumeration value="days" />
<xs:enumeration value="hours" />
<xs:enumeration value="minutes" />
<xs:enumeration value="seconds" />
<xs:enumeration value="metersPerSecond" />
<xs:enumeration value="kilometersPerHour" />
<xs:enumeration value="feetPerSecond" />
<xs:enumeration value="feetPerMinute" />
<xs:enumeration value="milesPerHour" />
<xs:enumeration value="cubicFeet" />
<xs:enumeration value="cubicMeters" />
<xs:enumeration value="imperialGallons" />
<xs:enumeration value="liters" />
<xs:enumeration value="usGallons" />
<xs:enumeration value="cubicFeetPerMinute" />
<xs:enumeration value="cubicMetersPerSecond" />
<xs:enumeration value="imperialGallonsPerMinute" />
<xs:enumeration value="litersPerSecond" />
<xs:enumeration value="litersPerMinute" />
<xs:enumeration value="usGallonsPerMinute" />
<xs:enumeration value="degreesAngular" />
<xs:enumeration value="degreesCelsiusPerHour" />
<xs:enumeration value="degreesCelsiusPerMinute" />
<xs:enumeration value="degreesFahrenheitPerHour" />
<xs:enumeration value="degreesFahrenheitPerMinute" />
<xs:enumeration value="noUnits" />
<xs:enumeration value="partsPerMillion" />
<xs:enumeration value="partsPerBillion" />
<xs:enumeration value="percent" />
<xs:enumeration value="percentPerSecond" />
<xs:enumeration value="perMinute" />
<xs:enumeration value="perSecond" />
<xs:enumeration value="psiPerDegreeFahrenheit" />
<xs:enumeration value="radians" />
<xs:enumeration value="revolutionsPerMinute" />
<xs:enumeration value="currency1" />
<xs:enumeration value="currency2" />
<xs:enumeration value="currency3" />
<xs:enumeration value="currency4" />
<xs:enumeration value="currency5" />
<xs:enumeration value="currency6" />
<xs:enumeration value="currency7" />
<xs:enumeration value="currency8" />
<xs:enumeration value="currency9" />
<xs:enumeration value="currency10" />
<xs:enumeration value="squareInches" />
<xs:enumeration value="squareCentimeters" />
<xs:enumeration value="btusPerPound" />
<xs:enumeration value="centimeters" />
<xs:enumeration value="poundsMassPerSecond" />
<xs:enumeration value="deltaDegreesFahrenheit" />
<xs:enumeration value="deltaDegreesKelvin" />
<xs:enumeration value="kilohms" />
<xs:enumeration value="megohms" />
<xs:enumeration value="millivolts" />
<xs:enumeration value="kilojoulesPerKilogram" />
<xs:enumeration value="megajoules" />
<xs:enumeration value="joulesPerDegreeKelvin" />
<xs:enumeration value="joulesPerKilogramDegreeKelvin" />
<xs:enumeration value="kilohertz" />
<xs:enumeration value="megahertz" />
<xs:enumeration value="perHour" />
<xs:enumeration value="milliwatts" />
<xs:enumeration value="hectopascals" />
<xs:enumeration value="millibars" />
<xs:enumeration value="cubicMetersPerHour" />
<xs:enumeration value="litersPerHour" />
<xs:enumeration value="kilowattHoursPerSquareMeter" />
<xs:enumeration value="kilowattHoursPerSquareFoot" />
<xs:enumeration value="megajoulesPerSquareMeter" />
```

```

<xs:enumeration value="megajoulesPerSquareFoot" />
<xs:enumeration value="wattsPerSquareMeterDegreeKelvin" />
<xs:enumeration value="cubicFeetPerSecond" />
<xs:enumeration value="percentObscurationPerFoot" />
<xs:enumeration value="percentObscurationPerMeter" />
<xs:enumeration value="milliohms" />
<xs:enumeration value="megawattHours" />
<xs:enumeration value="kiloBtus" />
<xs:enumeration value="megaBtus" />
<xs:enumeration value="kilojoulesPerKilogramDryAir" />
<xs:enumeration value="megajoulesPerKilogramDryAir" />
<xs:enumeration value="kilojoulesPerDegreeKelvin" />
<xs:enumeration value="megajoulesPerDegreeKelvin" />
<xs:enumeration value="newton" />
<xs:enumeration value="gramsPerSecond" />
<xs:enumeration value="gramsPerMinute" />
<xs:enumeration value="tonsPerHour" />
<xs:enumeration value="kiloBtusPerHour" />
<xs:enumeration value="hundredthsSeconds" />
<xs:enumeration value="milliseconds" />
<xs:enumeration value="newtonMeters" />
<xs:enumeration value="illimetersPerSecond" />
<xs:enumeration value="millimetersPerMinute" />
<xs:enumeration value="metersPerMinute" />
<xs:enumeration value="metersPerHour" />
<xs:enumeration value="cubicMetersPerMinute" />
<xs:enumeration value="metersPerSecondPerSecond" />
<xs:enumeration value="amperesPerMeter" />
<xs:enumeration value="amperesPerSquareMeter" />
<xs:enumeration value="ampereSquareMeters" />
<xs:enumeration value="farads" />
<xs:enumeration value="henrys" />
<xs:enumeration value="ohmMeters" />
<xs:enumeration value="siemens" />
<xs:enumeration value="siemensPerMeter" />
<xs:enumeration value="teslas" />
<xs:enumeration value="voltsPerDegreeKelvin" />
<xs:enumeration value="voltsPerMeter" />
<xs:enumeration value="webers" />
<xs:enumeration value="candelas" />
<xs:enumeration value="candelasPerSquareMeter" />
<xs:enumeration value="degreesKelvinPerHour" />
<xs:enumeration value="degreesKelvinPerMinute" />
<xs:enumeration value="jouleSeconds" />
<xs:enumeration value="radiansPerSecond" />
<xs:enumeration value="squareMetersPerNewton" />
<xs:enumeration value="kilogramsPerCubicMeter" />
<xs:enumeration value="newtonSeconds" />
<xs:enumeration value="newtonsPerMeter" />
<xs:enumeration value="wattsPerMeterPerDegreeKelvin" />
<xs:enumeration value="microsiemens" />
<xs:enumeration value="cubicFeetPerHour" />
<xs:enumeration value="usGallonsPerHour" />
<xs:enumeration value="kilometers" />
<xs:enumeration value="micrometers" />
<xs:enumeration value="grams" />
<xs:enumeration value="milligrams" />
<xs:enumeration value="milliliters" />
<xs:enumeration value="millilitersPerSecond" />
<xs:enumeration value="decibels" />
<xs:enumeration value="decibelsMillivolt" />
<xs:enumeration value="decibelsVolt" />
<xs:enumeration value="millisiemens" />
<xs:enumeration value="wattHoursReactive" />
<xs:enumeration value="kilowattHoursReactive" />
<xs:enumeration value="megawattHoursReactive" />
<xs:enumeration value="millimetersOfWater" />
<xs:enumeration value="perMille" />
<xs:enumeration value="gramsPerGram" />
<xs:enumeration value="kilogramsPerKilogram" />
<xs:enumeration value="gramsPerKilogram" />

```

```

<xs:enumeration value="milligramsPerGram" />
<xs:enumeration value="milligramsPerKilogram" />
<xs:enumeration value="gramsPerMilliliter" />
<xs:enumeration value="gramsPerLiter" />
<xs:enumeration value="milligramsPerLiter" />
<xs:enumeration value="microgramsPerLiter" />
<xs:enumeration value="gramsPerCubicMeter" />
<xs:enumeration value="milligramsPerCubicMeter" />
<xs:enumeration value="microgramsPerCubicMeter" />
<xs:enumeration value="nanogramsPerCubicMeter" />
<xs:enumeration value="gramsPerCubicCentimeter" />
<xs:enumeration value="becquerels" />
<xs:enumeration value="kilobecquerels" />
<xs:enumeration value="megabecquerels" />
<xs:enumeration value="gray" />
<xs:enumeration value="milligray" />
<xs:enumeration value="microgray" />
<xs:enumeration value="sieverts" />
<xs:enumeration value="millisieverts" />
<xs:enumeration value="microsieverts" />
<xs:enumeration value="microsievertsPerHour" />
<xs:enumeration value="decibelsA" />
<xs:enumeration value="nephelometricTurbidityUnit" />
<xs:enumeration value="pH" />
<xs:enumeration value="gramsPerSquareMeter" />
<xs:enumeration value="minutesPerDegreeKelvin" />
<xs:enumeration value="ohmMeterSquaredPerMeter" />
<xs:enumeration value="ampereSeconds" />
<xs:enumeration value="voltAmpereHours" />
<xs:enumeration value="kilovoltAmpereHours" />
<xs:enumeration value="megavoltAmpereHours" />
<xs:enumeration value="voltAmpereHoursReactive" />
<xs:enumeration value="kilovoltAmpereHoursReactive" />
<xs:enumeration value="megavoltAmpereHoursReactive" />
<xs:enumeration value="voltSquareHours" />
<xs:enumeration value="ampereSquareHours" />
<xs:enumeration value="joulePerHours" />
<xs:enumeration value="cubicFeetPerDay" />
<xs:enumeration value="cubicMetersPerDay" />
<xs:enumeration value="wattHoursPerCubicMeter" />
<xs:enumeration value="joulesPerCubicMeter" />
<xs:enumeration value="molePercent" />
<xs:enumeration value="pascalSeconds" />
<xs:enumeration value="millionStandardCubicFeetPerMinute" />
<xs:enumeration value="standardCubicFeetPerDay" />
<xs:enumeration value="millionStandardCubicFeetPerDay" />
<xs:enumeration value="thousandCubicFeetPerDay" />
<xs:enumeration value="thousandStandardCubicFeetPerDay" />
<xs:enumeration value="poundsMassPerDay" />
<xs:enumeration value="millirems" />
<xs:enumeration value="milliremsPerHour" />
</xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<BACnetEngineeringUnits>squareMeters</BACnetEngineeringUnits>
```

BACnetEscalatorFault

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetEscalatorFault">
  <xs:restriction base="xs:string">
    <xs:enumeration value="controllerFault" />
    <xs:enumeration value="driveAndMotorFault" />
    <xs:enumeration value="mechanicalComponentFault" />
    <xs:enumeration value="overspeedFault" />
    <xs:enumeration value="powerSupplyFault" />
    <xs:enumeration value="safetyDeviceFault" />
    <xs:enumeration value="controllerSupplyFault" />
    <xs:enumeration value="driveTemperatureExceeded" />
    <xs:enumeration value="combPlateFault" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetEscalatorFault>controllerFault</BACnetEscalatorFault>
```

BACnetEscalatorMode

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetEscalatorMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="unknown" />
    <xs:enumeration value="stop" />
    <xs:enumeration value="up" />
    <xs:enumeration value="down" />
    <xs:enumeration value="inspection" />
    <xs:enumeration value="outOfService" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetEscalatorMode>unknown</BACnetEscalatorMode>
```

BACnetEscalatorOperationDirection

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetEscalatorOperationDirection">
  <xs:restriction base="xs:string">
    <xs:enumeration value="unknown" />
    <xs:enumeration value="stopped" />
    <xs:enumeration value="upRatedSpeed" />
    <xs:enumeration value="upReducedSpeed" />
    <xs:enumeration value="downRatedSpeed" />
    <xs:enumeration value="downReducedSpeed" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetEscalatorOperationDirection>unknown</BACnetEscalatorOperationDirection>
```

BACnetEventLogRecord

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetEventLogRecord">
  <xs:sequence>
    <xs:element name="Timestamp" type="BACnetDateTime" />
    <xs:element name="LogDatum" type="BACnetEventLogRecord_LogDatum" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetEventLogRecord>
  <Timestamp>
    <Date>2017-12-20</Date>
    <Time>22:39:15.980</Time>
  </Timestamp>
  <LogDatum>
    <LogStatus>
      <logDisabled>true</logDisabled>
      <logInterrupted>true</logInterrupted>
    </LogStatus>
  </LogDatum>
</BACnetEventLogRecord>
```

BACnetEventLogRecord_LogDatum

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetEventLogRecord_LogDatum">
  <xs:sequence>
    <xs:choice>
      <xs:element name="LogStatus" type="BACnetLogStatus" />
      <xs:element name="Notification" type="ConfirmedEventNotificationRequest" />
      <xs:element name="TimeChange" type="REAL" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetEventLogRecord_LogDatum>
  <LogStatus>
    <logDisabled>true</logDisabled>
    <logInterrupted>true</logInterrupted>
  </LogStatus>
</BACnetEventLogRecord_LogDatum>
```

Example 2:


```

<BACnetEventLogRecord_LogDatum>
  <Notification>
    <ProcessIdentifier>2345</ProcessIdentifier>
    <InitiatingDeviceIdentifier>device-502</InitiatingDeviceIdentifier>
    <EventObjectIdentifier>device-503</EventObjectIdentifier>
    <Timestamp>
      <Time>22:39:15.980</Time>
    </Timestamp>
    <NotificationClass>1238</NotificationClass>
    <Priority>38</Priority>
    <EventType>changeOfLifeSafety</EventType>
    <MessageText>Abcde</MessageText>
    <NotifyType>event</NotifyType>
    <AckRequired>true</AckRequired>
    <FromState>offnormal</FromState>
    <ToState>highLimit</ToState>
    <EventValues>
      <ChangeOfBitstring>
        <ReferencedBitstring>1</ReferencedBitstring>
        <StatusFlags>
          <inAlarm>true</inAlarm>
          <overridden>true</overridden>
        </StatusFlags>
      </ChangeOfBitstring>
    </EventValues>
  </Notification>
</BACnetEventLogRecord_LogDatum>

```

Example 3:

```

<BACnetEventLogRecord_LogDatum>
  <TimeChange>1244.567749</TimeChange>
</BACnetEventLogRecord_LogDatum>

```

BACnetEventNotificationSubscription

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetEventNotificationSubscription">
  <xs:sequence>
    <xs:element name="Recipient" type="BACnetRecipient" />
    <xs:element name="ProcessIdentifier" type="Unsigned32" />
    <xs:element name="IssueConfirmedNotifications" type="Boolean" />
    <xs:element name="TimeRemaining" type="Unsigned" />
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```

<BACnetEventNotificationSubscription>
  <Recipient>
    <Device>device-501</Device>
  </Recipient>
  <ProcessIdentifier>2346</ProcessIdentifier>
  <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
  <TimeRemaining>1236</TimeRemaining>
</BACnetEventNotificationSubscription>

```

BACnetEventParameter_AccessEvent

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_AccessEvent">
    <xs:sequence>
      <xs:element name="ListOfAccessEvents" type="SequenceOfBACnetAccessEvent" />
      <xs:element name="AccessEventTimeReference" type="BACnetDeviceObjectPropertyReference" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_AccessEvent" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_AccessEvent>
    <ListOfAccessEvents />
    <AccessEventTimeReference>
      <ObjectIdentifier>device-501</ObjectIdentifier>
      <PropertyIdentifier>ackRequired</PropertyIdentifier>
      <PropertyArrayIndex>1236</PropertyArrayIndex>
      <DeviceIdentifier>device-504</DeviceIdentifier>
    </AccessEventTimeReference>
  </BACnetEventParameter_AccessEvent>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_AccessEvent>
    <ListOfAccessEvents>
      <BACnetAccessEvent>duress</BACnetAccessEvent>
      <BACnetAccessEvent>trace</BACnetAccessEvent>
      <BACnetAccessEvent>lockoutMaxAttempts</BACnetAccessEvent>
    </ListOfAccessEvents>
    <AccessEventTimeReference>
      <ObjectIdentifier>device-508</ObjectIdentifier>
      <PropertyIdentifier>all</PropertyIdentifier>
      <PropertyArrayIndex>1243</PropertyArrayIndex>
      <DeviceIdentifier>device-511</DeviceIdentifier>
    </AccessEventTimeReference>
  </BACnetEventParameter_AccessEvent>
</Value>
```

BACnetEventParameter_BufferReady

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_BufferReady">
    <xs:sequence>
      <xs:element name="NotificationThreshold" type="Unsigned" />
      <xs:element name="PreviousNotificationCount" type="Unsigned32" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_BufferReady" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetEventParameter_BufferReady>
    <NotificationThreshold>1234</NotificationThreshold>
    <PreviousNotificationCount>2346</PreviousNotificationCount>
  </BACnetEventParameter_BufferReady>
</Value>
```

BACnetEventParameter_ChangeOfBitstring

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_ChangeOfBitstring">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="Bitmask" type="BitString" />
      <xs:element name="ListOfBitstringValue" type="SequenceOfBitString" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_ChangeOfBitstring" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfBitstring>
    <TimeDelay>1234</TimeDelay>
    <Bitmask>101</Bitmask>
    <ListOfBitstringValue />
  </BACnetEventParameter_ChangeOfBitstring>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfBitstring>
    <TimeDelay>1235</TimeDelay>
    <Bitmask>101</Bitmask>
    <ListOfBitstringValues>
      <BitString>101</BitString>
      <BitString>1</BitString>
      <BitString>1</BitString>
    </ListOfBitstringValues>
  </BACnetEventParameter_ChangeOfBitstring>
</Value>
```

BACnetEventParameter_ChangeOfCharacterstring

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_ChangeOfCharacterstring">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="ListOfAlarmValues" type="SequenceOfCharacterString" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_ChangeOfCharacterstring" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfCharacterstring>
    <TimeDelay>1234</TimeDelay>
    <ListOfAlarmValues />
  </BACnetEventParameter_ChangeOfCharacterstring>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfCharacterstring>
    <TimeDelay>1235</TimeDelay>
    <ListOfAlarmValues>
      <CharacterString>Abcde</CharacterString>
      <CharacterString>Abcde</CharacterString>
      <CharacterString>Abcde</CharacterString>
    </ListOfAlarmValues>
  </BACnetEventParameter_ChangeOfCharacterstring>
</Value>
```

BACnetEventParameter_ChangeOfDiscreteValue

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_ChangeOfDiscreteValue">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_ChangeOfDiscreteValue" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfDiscreteValue>
    <TimeDelay>1234</TimeDelay>
  </BACnetEventParameter_ChangeOfDiscreteValue>
</Value>
```

BACnetEventParameter_ChangeOfLifeSafety

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_ChangeOfLifeSafety">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="ListOfLifeSafetyAlarmValues" type="SequenceOfBACnetLifeSafetyState" />
      <xs:element name="ListOfAlarmValues" type="SequenceOfBACnetLifeSafetyState" />
      <xs:element name="ModePropertyReference" type="BACnetDeviceObjectPropertyReference" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_ChangeOfLifeSafety" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfLifeSafety>
    <TimeDelay>1234</TimeDelay>
    <ListOfLifeSafetyAlarmValues />
    <ListOfAlarmValues />
    <ModePropertyReference>
      <ObjectIdentifier>device-502</ObjectIdentifier>
      <PropertyIdentifier>action</PropertyIdentifier>
      <PropertyArrayIndex>1237</PropertyArrayIndex>
      <DeviceIdentifier>device-505</DeviceIdentifier>
    </ModePropertyReference>
  </BACnetEventParameter_ChangeOfLifeSafety>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfLifeSafety>
    <TimeDelay>1239</TimeDelay>
    <ListOfLifeSafetyAlarmValues />
    <ListOfAlarmValues>
      <BACnetLifeSafetyState>notReady</BACnetLifeSafetyState>
      <BACnetLifeSafetyState>active</BACnetLifeSafetyState>
      <BACnetLifeSafetyState>tamper</BACnetLifeSafetyState>
    </ListOfAlarmValues>
    <ModePropertyReference>
      <ObjectIdentifier>device-510</ObjectIdentifier>
      <PropertyIdentifier>apduSegmentTimeout</PropertyIdentifier>
      <PropertyArrayIndex>1245</PropertyArrayIndex>
      <DeviceIdentifier>device-513</DeviceIdentifier>
    </ModePropertyReference>
  </BACnetEventParameter_ChangeOfLifeSafety>
</Value>
```

Example 3:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfLifeSafety>
    <TimeDelay>1247</TimeDelay>
    <ListOfLifeSafetyAlarmValues>
      <BACnetLifeSafetyState>duress</BACnetLifeSafetyState>
      <BACnetLifeSafetyState>tamperAlarm</BACnetLifeSafetyState>
      <BACnetLifeSafetyState>abnormal</BACnetLifeSafetyState>
    </ListOfLifeSafetyAlarmValues>
    <ListOfAlarmValues />
    <ModePropertyReference>
      <ObjectIdentifier>device-518</ObjectIdentifier>
      <PropertyIdentifier>controlledVariableReference</PropertyIdentifier>
      <PropertyArrayIndex>1253</PropertyArrayIndex>
      <DeviceIdentifier>device-521</DeviceIdentifier>
    </ModePropertyReference>
  </BACnetEventParameter_ChangeOfLifeSafety>
</Value>
```

Example 4:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfLifeSafety>
    <TimeDelay>1255</TimeDelay>
    <ListOfLifeSafetyAlarmValues>
      <BACnetLifeSafetyState>supervisory</BACnetLifeSafetyState>
      <BACnetLifeSafetyState>testSupervisory</BACnetLifeSafetyState>
      <BACnetLifeSafetyState>quiet</BACnetLifeSafetyState>
    </ListOfLifeSafetyAlarmValues>
    <ListOfAlarmValues>
      <BACnetLifeSafetyState>preAlarm</BACnetLifeSafetyState>
      <BACnetLifeSafetyState>alarm</BACnetLifeSafetyState>
      <BACnetLifeSafetyState>fault</BACnetLifeSafetyState>
    </ListOfAlarmValues>
    <ModePropertyReference>
      <ObjectIdentifier>device-529</ObjectIdentifier>
      <PropertyIdentifier>deviceAddressBinding</PropertyIdentifier>
      <PropertyArrayIndex>1264</PropertyArrayIndex>
      <DeviceIdentifier>device-532</DeviceIdentifier>
    </ModePropertyReference>
  </BACnetEventParameter_ChangeOfLifeSafety>
</Value>
```

BACnetEventParameter_ChangeOfState

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_ChangeOfState">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="ListOfValues" type="SequenceOfBACnetPropertyStates" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_ChangeOfState" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfState>
    <TimeDelay>1234</TimeDelay>
    <ListOfValues />
  </BACnetEventParameter_ChangeOfState>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfState>
    <TimeDelay>1235</TimeDelay>
    <ListOfValues>
      <BACnetPropertyStates>
        <BooleanValue>true</BooleanValue>
      </BACnetPropertyStates>
      <BACnetPropertyStates>
        <BooleanValue>true</BooleanValue>
      </BACnetPropertyStates>
      <BACnetPropertyStates>
        <BooleanValue>true</BooleanValue>
      </BACnetPropertyStates>
    </ListOfValues>
  </BACnetEventParameter_ChangeOfState>
</Value>
```

BACnetEventParameter_ChangeOfStatusFlags

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_ChangeOfStatusFlags">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="SelectedFlags" type="BACnetStatusFlags" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_ChangeOfStatusFlags" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfStatusFlags>
    <TimeDelay>1234</TimeDelay>
    <SelectedFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </SelectedFlags>
  </BACnetEventParameter_ChangeOfStatusFlags>
</Value>
```

BACnetEventParameter_ChangeOfTimer

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_ChangeOfTimer">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="AlarmValues" type="SequenceOfBACnetTimerState" />
      <xs:element name="UpdateTimeReference" type="BACnetDeviceObjectPropertyReference" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_ChangeOfTimer" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfTimer>
    <TimeDelay>1234</TimeDelay>
    <AlarmValues />
    <UpdateTimeReference>
      <ObjectIdentifier>device-502</ObjectIdentifier>
      <PropertyIdentifier>action</PropertyIdentifier>
      <PropertyArrayIndex>1237</PropertyArrayIndex>
      <DeviceIdentifier>device-505</DeviceIdentifier>
    </UpdateTimeReference>
  </BACnetEventParameter_ChangeOfTimer>
</Value>
```

Example 2:


```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfTimer>
    <TimeDelay>1239</TimeDelay>
    <AlarmValues>
      <BACnetTimerState>idle</BACnetTimerState>
      <BACnetTimerState>running</BACnetTimerState>
      <BACnetTimerState>expired</BACnetTimerState>
    </AlarmValues>
    <UpdateTimeReference>
      <ObjectIdentifier>device-510</ObjectIdentifier>
      <PropertyIdentifier>apduSegmentTimeout</PropertyIdentifier>
      <PropertyArrayIndex>1245</PropertyArrayIndex>
      <DeviceIdentifier>device-513</DeviceIdentifier>
    </UpdateTimeReference>
  </BACnetEventParameter_ChangeOfTimer>
</Value>
```

BACnetEventParameter_ChangeOfValue

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
  xmlns:msns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_ChangeOfValue">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="CovCriteria" type="BACnetEventParameter_ChangeOfValue_CovCriteria" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_ChangeOfValue" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_ChangeOfValue>
    <TimeDelay>1234</TimeDelay>
    <CovCriteria>
      <Bitmask>1</Bitmask>
    </CovCriteria>
  </BACnetEventParameter_ChangeOfValue>
</Value>
```

BACnetEventParameter_ChangeOfValue_CovCriteria

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetEventParameter_ChangeOfValue_CovCriteria">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Bitmask" type="BitString" />
      <xs:element name="ReferencedPropertyIncrement" type="REAL" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetEventParameter_ChangeOfValue_CovCriteria>
  <Bitmask>1</Bitmask>
</BACnetEventParameter_ChangeOfValue_CovCriteria>
```

Example 2:

```
<BACnetEventParameter_ChangeOfValue_CovCriteria>
  <ReferencedPropertyIncrement>1234.567749</ReferencedPropertyIncrement>
</BACnetEventParameter_ChangeOfValue_CovCriteria>
```

BACnetEventParameter_CommandFailure

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_CommandFailure">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="FeedbackPropertyReference" type="BACnetDeviceObjectPropertyReference" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_CommandFailure" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetEventParameter_CommandFailure>
    <TimeDelay>1234</TimeDelay>
    <FeedbackPropertyReference>
      <ObjectIdentifier>device-502</ObjectIdentifier>
      <PropertyIdentifier>action</PropertyIdentifier>
      <PropertyArrayIndex>1237</PropertyArrayIndex>
      <DeviceIdentifier>device-505</DeviceIdentifier>
    </FeedbackPropertyReference>
  </BACnetEventParameter_CommandFailure>
</Value>
```

BACnetEventParameter_DoubleOutOfRange

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_DoubleOutOfRange">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="LowLimit" type="Double" />
      <xs:element name="HighLimit" type="Double" />
      <xs:element name="Deadband" type="Double" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_DoubleOutOfRange" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_DoubleOutOfRange>
    <TimeDelay>1234</TimeDelay>
    <LowLimit>123457.789012</LowLimit>
    <HighLimit>123458.789012</HighLimit>
    <Deadband>123459.789012</Deadband>
  </BACnetEventParameter_DoubleOutOfRange>
</Value>
```

BACnetEventParameter_Extended

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_Extended">
    <xs:sequence>
      <xs:element name="VendorId" type="Unsigned16" />
      <xs:element name="ExtendedEventType" type="Unsigned" />
      <xs:element name="Parameters" type="SequenceOfBACnetEventParameter_Extended_Unnamed" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_Extended" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_Extended>
    <VendorId>7634</VendorId>
    <ExtendedEventType>1235</ExtendedEventType>
    <Parameters />
  </BACnetEventParameter_Extended>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetEventParameter_Extended>
    <VendorId>7636</VendorId>
    <ExtendedEventType>1237</ExtendedEventType>
    <Parameters>
      <Unnamed>
        <Null />
      </Unnamed>
      <Unnamed>
        <Null />
      </Unnamed>
      <Unnamed>
        <Null />
      </Unnamed>
    </Parameters>
  </BACnetEventParameter_Extended>
</Value>
```

BACnetEventParameter_Extended_Unnamed

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetEventParameter_Extended_Unnamed">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Null" type="Null" />
      <xs:element name="Real" type="REAL" />
      <xs:element name="Unsigned" type="Unsigned" />
      <xs:element name="Boolean" type="Boolean" />
      <xs:element name="Integer" type="INTEGER" />
      <xs:element name="Double" type="Double" />
      <xs:element name="Octetstring" type="OctetString" />
      <xs:element name="Characterstring" type="CharacterString" />
      <xs:element name="Bitstring" type="BitString" />
      <xs:element name="Enumerated" type="Enumerated" />
      <xs:element name="Date" type="Date" />
      <xs:element name="Time" type="Time" />
      <xs:element name="Objectidentifier" type="BACnetObjectIdentifier" />
      <xs:element name="Reference" type="BACnetDeviceObjectPropertyReference" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetEventParameter_Extended_Unnamed>
  <Null />
</BACnetEventParameter_Extended_Unnamed>
```

Example 2:

```
<BACnetEventParameter_Extended_Unnamed>
  <Real>1234.567749</Real>
</BACnetEventParameter_Extended_Unnamed>
```

Example 3:

```
<BACnetEventParameter_Extended_Unnamed>
  <Unsigned>1235</Unsigned>
</BACnetEventParameter_Extended_Unnamed>
```

Example 4:

```
<BACnetEventParameter_Extended_Unnamed>
  <Boolean>true</Boolean>
</BACnetEventParameter_Extended_Unnamed>
```

Example 5:

```
<BACnetEventParameter_Extended_Unnamed>
  <Integer>-798</Integer>
</BACnetEventParameter_Extended_Unnamed>
```

Example 6:

```
<BACnetEventParameter_Extended_Unnamed>
  <Double>123459.789012</Double>
</BACnetEventParameter_Extended_Unnamed>
```

Example 7:

```
<BACnetEventParameter_Extended_Unnamed>
  <Octetstring>FF</Octetstring>
</BACnetEventParameter_Extended_Unnamed>
```

Example 8:

```
<BACnetEventParameter_Extended_Unnamed>
  <Characterstring>Abcde</Characterstring>
</BACnetEventParameter_Extended_Unnamed>
```

Example 9:

```
<BACnetEventParameter_Extended_Unnamed>
  <Bitstring>1</Bitstring>
</BACnetEventParameter_Extended_Unnamed>
```

Example 10:

```
<BACnetEventParameter_Extended_Unnamed>
  <Enumerated>4</Enumerated>
</BACnetEventParameter_Extended_Unnamed>
```

Example 11:

```
<BACnetEventParameter_Extended_Unnamed>
  <Date>2022-12-20</Date>
</BACnetEventParameter_Extended_Unnamed>
```

Example 12:

```
<BACnetEventParameter_Extended_Unnamed>
  <Time>22:39:15.980</Time>
</BACnetEventParameter_Extended_Unnamed>
```

Example 13:

```
<BACnetEventParameter_Extended_Unnamed>
  <ObjectIdentifier>device-508</ObjectIdentifier>
</BACnetEventParameter_Extended_Unnamed>
```

Example 14:

```
<BACnetEventParameter_Extended_Unnamed>
  <Reference>
    <ObjectIdentifier>device-509</ObjectIdentifier>
    <PropertyIdentifier>allWritesSuccessful</PropertyIdentifier>
    <PropertyArrayIndex>1244</PropertyArrayIndex>
    <DeviceIdentifier>device-512</DeviceIdentifier>
  </Reference>
</BACnetEventParameter_Extended_Unnamed>
```

BACnetEventParameter_FloatingLimit

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_FloatingLimit">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="SetpointReference" type="BACnetDeviceObjectPropertyReference" />
      <xs:element name="LowDiffLimit" type="REAL" />
      <xs:element name="HighDiffLimit" type="REAL" />
      <xs:element name="Deadband" type="REAL" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_FloatingLimit" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetEventParameter_FloatingLimit>
    <TimeDelay>1234</TimeDelay>
    <SetpointReference>
      <ObjectIdentifier>device-502</ObjectIdentifier>
      <PropertyIdentifier>action</PropertyIdentifier>
      <PropertyArrayIndex>1237</PropertyArrayIndex>
      <DeviceIdentifier>device-505</DeviceIdentifier>
    </SetpointReference>
    <LowDiffLimit>1239.567749</LowDiffLimit>
    <HighDiffLimit>1240.567749</HighDiffLimit>
    <Deadband>1241.567749</Deadband>
  </BACnetEventParameter_FloatingLimit>
</Value>
```

BACnetEventParameter_OutOfRange

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_OutOfRange">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="LowLimit" type="REAL" />
      <xs:element name="HighLimit" type="REAL" />
      <xs:element name="Deadband" type="REAL" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_OutOfRange" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetEventParameter_OutOfRange>
    <TimeDelay>1234</TimeDelay>
    <LowLimit>1235.567749</LowLimit>
    <HighLimit>1236.567749</HighLimit>
    <Deadband>1237.567749</Deadband>
  </BACnetEventParameter_OutOfRange>
</Value>
```

BACnetEventParameter_SignedOutOfRange

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_SignedOutOfRange">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="LowLimit" type="INTEGER" />
      <xs:element name="HighLimit" type="INTEGER" />
      <xs:element name="Deadband" type="Unsigned" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_SignedOutOfRange" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetEventParameter_SignedOutOfRange>
    <TimeDelay>1234</TimeDelay>
    <LowLimit>-799</LowLimit>
    <HighLimit>-798</HighLimit>
    <Deadband>1237</Deadband>
  </BACnetEventParameter_SignedOutOfRange>
</Value>
```

BACnetEventParameter_UnsignedOutOfRange

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_UnsignedOutOfRange">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="LowLimit" type="Unsigned" />
      <xs:element name="HighLimit" type="Unsigned" />
      <xs:element name="Deadband" type="Unsigned" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_UnsignedOutOfRange" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetEventParameter_UnsignedOutOfRange>
    <TimeDelay>1234</TimeDelay>
    <LowLimit>1235</LowLimit>
    <HighLimit>1236</HighLimit>
    <Deadband>1237</Deadband>
  </BACnetEventParameter_UnsignedOutOfRange>
</Value>
```

BACnetEventParameter_UnsignedRange

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.eventParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetEventParameter_UnsignedRange">
    <xs:sequence>
      <xs:element name="TimeDelay" type="Unsigned" />
      <xs:element name="LowLimit" type="Unsigned" />
      <xs:element name="HighLimit" type="Unsigned" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetEventParameter_UnsignedRange" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetEventParameter_UnsignedRange>
    <TimeDelay>1234</TimeDelay>
    <LowLimit>1235</LowLimit>
    <HighLimit>1236</HighLimit>
  </BACnetEventParameter_UnsignedRange>
</Value>
```

BACnetEventState

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetEventState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="normal" />
    <xs:enumeration value="fault" />
    <xs:enumeration value="offnormal" />
    <xs:enumeration value="highLimit" />
    <xs:enumeration value="lowLimit" />
    <xs:enumeration value="lifeSafetyAlarm" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetEventState>normal</BACnetEventState>
```

BACnetEventTransitionBits

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetEventTransitionBits">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="toOffnormal" type="Boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="toFault" type="Boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="toNormal" type="Boolean" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetEventTransitionBits>
  <toOffnormal>true</toOffnormal>
  <toNormal>true</toNormal>
</BACnetEventTransitionBits>
```

BACnetEventType

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:simpleType name="BACnetEventType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="changeOfBitstring" />
    <xs:enumeration value="changeOfState" />
    <xs:enumeration value="changeOfValue" />
    <xs:enumeration value="commandFailure" />
    <xs:enumeration value="floatingLimit" />
    <xs:enumeration value="outOfRange" />
    <xs:enumeration value="changeOfLifeSafety" />
    <xs:enumeration value="extended" />
    <xs:enumeration value="bufferReady" />
    <xs:enumeration value="unsignedRange" />
    <xs:enumeration value="accessEvent" />
    <xs:enumeration value="doubleOutOfRange" />
    <xs:enumeration value="signedOutOfRange" />
    <xs:enumeration value="unsignedOutOfRange" />
    <xs:enumeration value="changeOfCharacterstring" />
    <xs:enumeration value="changeOfStatusFlags" />
    <xs:enumeration value="changeOfReliability" />
    <xs:enumeration value="none" />
    <xs:enumeration value="changeOfDiscreteValue" />
    <xs:enumeration value="changeOfTimer" />
  </xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<BACnetEventType>changeOfBitstring</BACnetEventType>
```

BACnetFDTEEntry

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetFDTEEntry">
  <xs:sequence>
    <xs:element name="BacnetipAddress" type="OctetString" />
    <xs:element name="TimeToLive" type="Unsigned16" />
    <xs:element name="RemainingTimeToLive" type="Unsigned16" />
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```

<BACnetFDTEEntry>
  <BacnetipAddress>00010203040506070809</BacnetipAddress>
  <TimeToLive>7634</TimeToLive>
  <RemainingTimeToLive>7635</RemainingTimeToLive>
</BACnetFDTEEntry>

```

BACnetFaultParameter_FaultCharacterstring

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.faultParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetFaultParameter_FaultCharacterstring">
    <xs:sequence>
      <xs:element name="ListOfFaultValues" type="SequenceOfCharacterString" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetFaultParameter_FaultCharacterstring" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetFaultParameter_FaultCharacterstring>
    <ListOfFaultValues />
  </BACnetFaultParameter_FaultCharacterstring>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetFaultParameter_FaultCharacterstring>
    <ListOfFaultValues>
      <CharacterString>Abcde</CharacterString>
      <CharacterString>Abcde</CharacterString>
      <CharacterString>Abcde</CharacterString>
    </ListOfFaultValues>
  </BACnetFaultParameter_FaultCharacterstring>
</Value>
```

BACnetFaultParameter_FaultExtended

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.faultParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetFaultParameter_FaultExtended">
    <xs:sequence>
      <xs:element name="VendorId" type="Unsigned16" />
      <xs:element name="ExtendedFaultType" type="Unsigned" />
      <xs:element name="Parameters" type="SequenceOfBACnetFaultParameter_FaultExtended_Unnamed" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetFaultParameter_FaultExtended" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetFaultParameter_FaultExtended>
    <VendorId>7634</VendorId>
    <ExtendedFaultType>1235</ExtendedFaultType>
    <Parameters />
  </BACnetFaultParameter_FaultExtended>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetFaultParameter_FaultExtended>
    <VendorId>7636</VendorId>
    <ExtendedFaultType>1237</ExtendedFaultType>
    <Parameters>
      <Unnamed>
        <Null />
      </Unnamed>
      <Unnamed>
        <Null />
      </Unnamed>
      <Unnamed>
        <Null />
      </Unnamed>
    </Parameters>
  </BACnetFaultParameter_FaultExtended>
</Value>
```

BACnetFaultParameter_FaultExtended_Unnamed

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetFaultParameter_FaultExtended_Unnamed">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Null" type="Null" />
      <xs:element name="Real" type="REAL" />
      <xs:element name="Unsigned" type="Unsigned" />
      <xs:element name="Boolean" type="Boolean" />
      <xs:element name="Integer" type="INTEGER" />
      <xs:element name="Double" type="Double" />
      <xs:element name="Octetstring" type="OctetString" />
      <xs:element name="Characterstring" type="CharacterString" />
      <xs:element name="Bitstring" type="BitString" />
      <xs:element name="Enumerated" type="Enumerated" />
      <xs:element name="Date" type="Date" />
      <xs:element name="Time" type="Time" />
      <xs:element name="Objectidentifier" type="BACnetObjectIdentifier" />
      <xs:element name="Reference" type="BACnetDeviceObjectPropertyReference" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <Null />
</BACnetFaultParameter_FaultExtended_Unnamed>
```

Example 2:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <Real>1234.567749</Real>
</BACnetFaultParameter_FaultExtended_Unnamed>
```

Example 3:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <Unsigned>1235</Unsigned>
</BACnetFaultParameter_FaultExtended_Unnamed>
```

Example 4:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <Boolean>true</Boolean>
</BACnetFaultParameter_FaultExtended_Unnamed>
```

Example 5:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <Integer>-798</Integer>
</BACnetFaultParameter_FaultExtended_Unnamed>
```

Example 6:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <Double>123459.789012</Double>
</BACnetFaultParameter_FaultExtended_Unnamed>
```

Example 7:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <Octetstring>FF</Octetstring>
</BACnetFaultParameter_FaultExtended_Unnamed>
```

Example 8:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <Characterstring>Abcde</Characterstring>
</BACnetFaultParameter_FaultExtended_Unnamed>
```

Example 9:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <Bitstring>1</Bitstring>
</BACnetFaultParameter_FaultExtended_Unnamed>
```

Example 10:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <Enumerated>4</Enumerated>
</BACnetFaultParameter_FaultExtended_Unnamed>
```

Example 11:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <Date>2022-12-20</Date>
</BACnetFaultParameter_FaultExtended_Unnamed>
```

Example 12:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <Time>22:39:15.980</Time>
</BACnetFaultParameter_FaultExtended_Unnamed>
```

Example 13:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <ObjectIdentifier>device-508</ObjectIdentifier>
</BACnetFaultParameter_FaultExtended_Unnamed>
```

Example 14:

```
<BACnetFaultParameter_FaultExtended_Unnamed>
  <Reference>
    <ObjectIdentifier>device-509</ObjectIdentifier>
    <PropertyIdentifier>allWritesSuccessful</PropertyIdentifier>
    <PropertyArrayIndex>1244</PropertyArrayIndex>
    <DeviceIdentifier>device-512</DeviceIdentifier>
  </Reference>
</BACnetFaultParameter_FaultExtended_Unnamed>
```

BACnetFaultParameter_FaultLifeSafety

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.faultParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetFaultParameter_FaultLifeSafety">
    <xs:sequence>
      <xs:element name="ListOfFaultValues" type="SequenceOfBACnetLifeSafetyState" />
      <xs:element name="ModePropertyReference" type="BACnetDeviceObjectPropertyReference" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetFaultParameter_FaultLifeSafety" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetFaultParameter_FaultLifeSafety>
    <ListOfFaultValues />
    <ModePropertyReference>
      <ObjectIdentifier>device-501</ObjectIdentifier>
      <PropertyIdentifier>ackRequired</PropertyIdentifier>
      <PropertyArrayIndex>1236</PropertyArrayIndex>
      <DeviceIdentifier>device-504</DeviceIdentifier>
    </ModePropertyReference>
  </BACnetFaultParameter_FaultLifeSafety>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetFaultParameter_FaultLifeSafety>
    <ListOfFaultValues>
      <BACnetLifeSafetyState>faultPreAlarm</BACnetLifeSafetyState>
      <BACnetLifeSafetyState>faultAlarm</BACnetLifeSafetyState>
      <BACnetLifeSafetyState>notReady</BACnetLifeSafetyState>
    </ListOfFaultValues>
    <ModePropertyReference>
      <ObjectIdentifier>device-508</ObjectIdentifier>
      <PropertyIdentifier>all</PropertyIdentifier>
      <PropertyArrayIndex>1243</PropertyArrayIndex>
      <DeviceIdentifier>device-511</DeviceIdentifier>
    </ModePropertyReference>
  </BACnetFaultParameter_FaultLifeSafety>
</Value>
```

BACnetFaultParameter_FaultListed

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.faultParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetFaultParameter_FaultListed">
    <xs:sequence>
      <xs:element name="FaultListReference" type="BACnetDeviceObjectPropertyReference" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetFaultParameter_FaultListed" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetFaultParameter_FaultListed>
    <FaultListReference>
      <ObjectIdentifier>device-501</ObjectIdentifier>
      <PropertyIdentifier>ackRequired</PropertyIdentifier>
      <PropertyArrayIndex>1236</PropertyArrayIndex>
      <DeviceIdentifier>device-504</DeviceIdentifier>
    </FaultListReference>
  </BACnetFaultParameter_FaultListed>
</Value>
```

BACnetFaultParameter_FaultOutOfRange

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.faultParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetFaultParameter_FaultOutOfRange">
    <xs:sequence>
      <xs:element name="MinNormalValue"
type="BACnetFaultParameter_FaultOutOfRange_MinNormalValue" />
      <xs:element name="MaxNormalValue"
type="BACnetFaultParameter_FaultOutOfRange_MaxNormalValue" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetFaultParameter_FaultOutOfRange" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetFaultParameter_FaultOutOfRange>
    <MinNormalValue>
      <Real>1234.567749</Real>
    </MinNormalValue>
    <MaxNormalValue>
      <Real>1235.567749</Real>
    </MaxNormalValue>
  </BACnetFaultParameter_FaultOutOfRange>
</Value>
```

BACnetFaultParameter_FaultOutOfRange_MaxNormalValue

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetFaultParameter_FaultOutOfRange_MaxNormalValue">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Real" type="REAL" />
      <xs:element name="Unsigned" type="Unsigned" />
      <xs:element name="Double" type="Double" />
      <xs:element name="Integer" type="INTEGER" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetFaultParameter_FaultOutOfRange_MaxNormalValue>
  <Real>1234.567749</Real>
</BACnetFaultParameter_FaultOutOfRange_MaxNormalValue>
```

Example 2:

```
<BACnetFaultParameter_FaultOutOfRange_MaxNormalValue>
  <Unsigned>1235</Unsigned>
</BACnetFaultParameter_FaultOutOfRange_MaxNormalValue>
```

Example 3:

```
<BACnetFaultParameter_FaultOutOfRange_MaxNormalValue>
  <Double>123458.789012</Double>
</BACnetFaultParameter_FaultOutOfRange_MaxNormalValue>
```

Example 4:

```
<BACnetFaultParameter_FaultOutOfRange_MaxNormalValue>
  <Integer>-797</Integer>
</BACnetFaultParameter_FaultOutOfRange_MaxNormalValue>
```

BACnetFaultParameter_FaultOutOfRange_MinNormalValue

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetFaultParameter_FaultOutOfRange_MinNormalValue">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Real" type="REAL" />
      <xs:element name="Unsigned" type="Unsigned" />
      <xs:element name="Double" type="Double" />
      <xs:element name="Integer" type="INTEGER" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetFaultParameter_FaultOutOfRange_MinNormalValue>
  <Real>1234.567749</Real>
</BACnetFaultParameter_FaultOutOfRange_MinNormalValue>
```

Example 2:


```
<BACnetFaultParameter_FaultOutOfRange_MinNormalValue>
  <Unsigned>1235</Unsigned>
</BACnetFaultParameter_FaultOutOfRange_MinNormalValue>
```

Example 3:

```
<BACnetFaultParameter_FaultOutOfRange_MinNormalValue>
  <Double>123458.789012</Double>
</BACnetFaultParameter_FaultOutOfRange_MinNormalValue>
```

Example 4:

```
<BACnetFaultParameter_FaultOutOfRange_MinNormalValue>
  <Integer>-797</Integer>
</BACnetFaultParameter_FaultOutOfRange_MinNormalValue>
```

BACnetFaultParameter_FaultState

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.faultParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetFaultParameter_FaultState">
    <xs:sequence>
      <xs:element name="ListOfFaultValues" type="SequenceOfBACnetPropertyStates" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetFaultParameter_FaultState" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetFaultParameter_FaultState>
    <ListOfFaultValues />
  </BACnetFaultParameter_FaultState>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetFaultParameter_FaultState>
    <ListOfFaultValues>
      <BACnetPropertyStates>
        <BooleanValue>true</BooleanValue>
      </BACnetPropertyStates>
      <BACnetPropertyStates>
        <BooleanValue>true</BooleanValue>
      </BACnetPropertyStates>
      <BACnetPropertyStates>
        <BooleanValue>true</BooleanValue>
      </BACnetPropertyStates>
    </ListOfFaultValues>
  </BACnetFaultParameter_FaultState>
</Value>
```

BACnetFaultParameter_FaultStatusFlags

This XML representation of complex data is used by the following objects/properties:

- **eventEnrollment.faultParameters**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetFaultParameter_FaultStatusFlags">
    <xs:sequence>
      <xs:element name="StatusFlagsReference" type="BACnetDeviceObjectPropertyReference" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetFaultParameter_FaultStatusFlags" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetFaultParameter_FaultStatusFlags>
    <StatusFlagsReference>
      <ObjectIdentifier>device-501</ObjectIdentifier>
      <PropertyIdentifier>ackRequired</PropertyIdentifier>
      <PropertyArrayIndex>1236</PropertyArrayIndex>
      <DeviceIdentifier>device-504</DeviceIdentifier>
    </StatusFlagsReference>
  </BACnetFaultParameter_FaultStatusFlags>
</Value>
```

BACnetFileAccessMethod

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetFileAccessMethod">
  <xs:restriction base="xs:string">
    <xs:enumeration value="recordAccess" />
    <xs:enumeration value="streamAccess" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetFileAccessMethod>recordAccess</BACnetFileAccessMethod>
```

BACnetHostAddress

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetHostAddress">
  <xs:sequence>
    <xs:choice>
      <xs:element name="None" type="Null" />
      <xs:element name="IpAddress" type="OctetString" />
      <xs:element name="Name" type="CharacterString" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetHostAddress>
  <None />
</BACnetHostAddress>
```

Example 2:

```
<BACnetHostAddress>
  <IpAddress>FF</IpAddress>
</BACnetHostAddress>
```

Example 3:

```
<BACnetHostAddress>
  <Name>Abcde</Name>
</BACnetHostAddress>
```

BACnetHostNPort

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetHostNPort">
  <xs:sequence>
    <xs:element name="Host" type="BACnetHostAddress" />
    <xs:element name="Port" type="Unsigned16" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetHostNPort>
  <Host>
    <None />
  </Host>
  <Port>7634</Port>
</BACnetHostNPort>
```

BACnetIPMode

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetIPMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="normal" />
    <xs:enumeration value="foreign" />
    <xs:enumeration value="bbmd" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetIPMode>normal</BACnetIPMode>
```

BACnetKeyIdentifier

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetKeyIdentifier">
  <xs:sequence>
    <xs:element name="Algorithm" type="Unsigned8" />
    <xs:element name="KeyId" type="Unsigned8" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetKeyIdentifier>
  <Algorithm>33</Algorithm>
  <KeyId>34</KeyId>
</BACnetKeyIdentifier>
```

BACnetLandingCallStatus

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetLandingCallStatus">
  <xs:sequence>
    <xs:element name="FloorNumber" type="Unsigned8" />
    <xs:element name="Command" type="BACnetLandingCallStatus_Command" />
    <xs:element minOccurs="0" name="FloorText" type="CharacterString" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetLandingCallStatus>
  <FloorNumber>33</FloorNumber>
  <Command>
    <Direction>none</Direction>
  </Command>
  <FloorText>Abcde</FloorText>
</BACnetLandingCallStatus>
```

Example 2:

```
<BACnetLandingCallStatus>
  <FloorNumber>35</FloorNumber>
  <Command>
    <Direction>up</Direction>
  </Command>
</BACnetLandingCallStatus>
```

BACnetLandingCallStatus_Command

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetLandingCallStatus_Command">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Direction" type="BACnetLiftCarDirection" />
      <xs:element name="Destination" type="Unsigned8" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetLandingCallStatus_Command>
  <Direction>unknown</Direction>
</BACnetLandingCallStatus_Command>
```

Example 2:

```
<BACnetLandingCallStatus_Command>
  <Destination>34</Destination>
</BACnetLandingCallStatus_Command>
```

BACnetLandingDoorStatus

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetLandingDoorStatus">
  <xs:sequence>
    <xs:element name="LandingDoors" type="BACnetLandingDoorStatus_LandingDoors" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetLandingDoorStatus>
  <LandingDoors />
</BACnetLandingDoorStatus>
```

Example 2:

```
<BACnetLandingDoorStatus>
  <LandingDoors>
    <LandingDoors_Item>
      <FloorNumber>33</FloorNumber>
      <DoorStatus>opened</DoorStatus>
    </LandingDoors_Item>
    <LandingDoors_Item>
      <FloorNumber>35</FloorNumber>
      <DoorStatus>doorFault</DoorStatus>
    </LandingDoors_Item>
    <LandingDoors_Item>
      <FloorNumber>37</FloorNumber>
      <DoorStatus>none</DoorStatus>
    </LandingDoors_Item>
  </LandingDoors>
</BACnetLandingDoorStatus>
```

BACnetLandingDoorStatus_LandingDoors

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetLandingDoorStatus_LandingDoors">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="LandingDoors_Item"
type="BACnetLandingDoorStatus_LandingDoors_Item" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<SequenceOfBACnetLandingDoorStatus_LandingDoors />
```

Example 2:

```
<SequenceOfBACnetLandingDoorStatus_LandingDoors>
  <LandingDoors />
  <LandingDoors>
    <LandingDoors_Item>
      <FloorNumber>33</FloorNumber>
      <DoorStatus>opened</DoorStatus>
    </LandingDoors_Item>
  </LandingDoors>
  <LandingDoors>
    <LandingDoors_Item>
      <FloorNumber>35</FloorNumber>
      <DoorStatus>doorFault</DoorStatus>
    </LandingDoors_Item>
  </LandingDoors>
</SequenceOfBACnetLandingDoorStatus_LandingDoors>
```

Example 3:

```
<SequenceOfBACnetLandingDoorStatus_LandingDoors>
  <LandingDoors>
    <LandingDoors_Item>
      <FloorNumber>37</FloorNumber>
      <DoorStatus>none</DoorStatus>
    </LandingDoors_Item>
    <LandingDoors_Item>
      <FloorNumber>39</FloorNumber>
      <DoorStatus>opening</DoorStatus>
    </LandingDoors_Item>
    <LandingDoors_Item>
      <FloorNumber>41</FloorNumber>
      <DoorStatus>limitedOpened</DoorStatus>
    </LandingDoors_Item>
  </LandingDoors>
  <LandingDoors>
    <LandingDoors_Item>
      <FloorNumber>43</FloorNumber>
      <DoorStatus>opened</DoorStatus>
    </LandingDoors_Item>
  </LandingDoors>
  <LandingDoors>
    <LandingDoors_Item>
      <FloorNumber>45</FloorNumber>
      <DoorStatus>doorFault</DoorStatus>
    </LandingDoors_Item>
  </LandingDoors>
</SequenceOfBACnetLandingDoorStatus_LandingDoors>
```

BACnetLandingDoorStatus_LandingDoors_Item

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetLandingDoorStatus_LandingDoors_Item">
  <xs:sequence>
    <xs:element name="FloorNumber" type="Unsigned8" />
    <xs:element name="DoorStatus" type="BACnetDoorStatus" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetLandingDoorStatus_LandingDoors_Item>
  <FloorNumber>33</FloorNumber>
  <DoorStatus>opened</DoorStatus>
</BACnetLandingDoorStatus_LandingDoors_Item>
```

BACnetLifeSafetyMode

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetLifeSafetyMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="off" />
    <xs:enumeration value="on" />
    <xs:enumeration value="test" />
    <xs:enumeration value="manned" />
    <xs:enumeration value="unmanned" />
    <xs:enumeration value="armed" />
    <xs:enumeration value="disarmed" />
    <xs:enumeration value="prearmed" />
    <xs:enumeration value="slow" />
    <xs:enumeration value="fast" />
    <xs:enumeration value="disconnected" />
    <xs:enumeration value="enabled" />
    <xs:enumeration value="disabled" />
    <xs:enumeration value="automaticReleaseDisabled" />
    <xs:enumeration value="default" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetLifeSafetyMode>off</BACnetLifeSafetyMode>
```

BACnetLifeSafetyOperation

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetLifeSafetyOperation">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none" />
    <xs:enumeration value="silence" />
    <xs:enumeration value="silenceAudible" />
    <xs:enumeration value="silenceVisual" />
    <xs:enumeration value="reset" />
    <xs:enumeration value="resetAlarm" />
    <xs:enumeration value="resetFault" />
    <xs:enumeration value="unsilence" />
    <xs:enumeration value="unsilenceAudible" />
    <xs:enumeration value="unsilenceVisual" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetLifeSafetyOperation>none</BACnetLifeSafetyOperation>
```

BACnetLifeSafetyState

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:simpleType name="BACnetLifeSafetyState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="quiet" />
    <xs:enumeration value="preAlarm" />
    <xs:enumeration value="alarm" />
    <xs:enumeration value="fault" />
    <xs:enumeration value="faultPreAlarm" />
    <xs:enumeration value="faultAlarm" />
    <xs:enumeration value="notReady" />
    <xs:enumeration value="active" />
    <xs:enumeration value="tamper" />
    <xs:enumeration value="testAlarm" />
    <xs:enumeration value="testActive" />
    <xs:enumeration value="testFault" />
    <xs:enumeration value="testFaultAlarm" />
    <xs:enumeration value="holdup" />
    <xs:enumeration value="duress" />
    <xs:enumeration value="tamperAlarm" />
    <xs:enumeration value="abnormal" />
    <xs:enumeration value="emergencyPower" />
    <xs:enumeration value="delayed" />
    <xs:enumeration value="blocked" />
    <xs:enumeration value="localAlarm" />
    <xs:enumeration value="generalAlarm" />
    <xs:enumeration value="supervisory" />
    <xs:enumeration value="testSupervisory" />
  </xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<BACnetLifeSafetyState>quiet</BACnetLifeSafetyState>
```

BACnetLiftCarCallList

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetLiftCarCallList">
  <xs:sequence>
    <xs:element name="FloorNumbers" type="SequenceOfUnsigned8" />
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```

<BACnetLiftCarCallList>
  <FloorNumbers />
</BACnetLiftCarCallList>

```

Example 2:

```

<BACnetLiftCarCallList>
  <FloorNumbers>
    <Unsigned8>33</Unsigned8>
    <Unsigned8>34</Unsigned8>
    <Unsigned8>35</Unsigned8>
  </FloorNumbers>
</BACnetLiftCarCallList>

```

BACnetLiftCarDirection

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema


```
<xs:simpleType name="BACnetLiftCarDirection">
  <xs:restriction base="xs:string">
    <xs:enumeration value="unknown" />
    <xs:enumeration value="none" />
    <xs:enumeration value="stopped" />
    <xs:enumeration value="up" />
    <xs:enumeration value="down" />
    <xs:enumeration value="upAndDown" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetLiftCarDirection>unknown</BACnetLiftCarDirection>
```

BACnetLiftCarDoorCommand

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetLiftCarDoorCommand">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none" />
    <xs:enumeration value="open" />
    <xs:enumeration value="close" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetLiftCarDoorCommand>none</BACnetLiftCarDoorCommand>
```

BACnetLiftCarDriveStatus

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetLiftCarDriveStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="unknown" />
    <xs:enumeration value="stationary" />
    <xs:enumeration value="braking" />
    <xs:enumeration value="accelerate" />
    <xs:enumeration value="decelerate" />
    <xs:enumeration value="ratedSpeed" />
    <xs:enumeration value="singleFloorJump" />
    <xs:enumeration value="twoFloorJump" />
    <xs:enumeration value="threeFloorJump" />
    <xs:enumeration value="multiFloorJump" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetLiftCarDriveStatus>unknown</BACnetLiftCarDriveStatus>
```

BACnetLiftCarMode

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:simpleType name="BACnetLiftCarMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="unknown" />
    <xs:enumeration value="normal" />
    <xs:enumeration value="vip" />
    <xs:enumeration value="homing" />
    <xs:enumeration value="parking" />
    <xs:enumeration value="attendantControl" />
    <xs:enumeration value="firefighterControl" />
    <xs:enumeration value="emergencyPower" />
    <xs:enumeration value="inspection" />
    <xs:enumeration value="cabinetRecall" />
    <xs:enumeration value="earthquakeOperation" />
    <xs:enumeration value="fireOperation" />
    <xs:enumeration value="outOfService" />
    <xs:enumeration value="occupantEvacuation" />
  </xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<BACnetLiftCarMode>unknown</BACnetLiftCarMode>
```

BACnetLiftFault

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:simpleType name="BACnetLiftFault">
  <xs:restriction base="xs:string">
    <xs:enumeration value="controllerFault" />
    <xs:enumeration value="driveAndMotorFault" />
    <xs:enumeration value="governorAndSafetyGearFault" />
    <xs:enumeration value="liftShaftDeviceFault" />
    <xs:enumeration value="powerSupplyFault" />
    <xs:enumeration value="safetyInterlockFault" />
    <xs:enumeration value="doorClosingFault" />
    <xs:enumeration value="doorOpeningFault" />
    <xs:enumeration value="carStoppedOutsideLandingZone" />
    <xs:enumeration value="callButtonStuck" />
    <xs:enumeration value="startFailure" />
    <xs:enumeration value="controllerSupplyFault" />
    <xs:enumeration value="selfTestFailure" />
    <xs:enumeration value="runtimeLimitExceeded" />
    <xs:enumeration value="positionLost" />
    <xs:enumeration value="driveTemperatureExceeded" />
    <xs:enumeration value="loadMeasurementFault" />
  </xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<BACnetLiftFault>controllerFault</BACnetLiftFault>
```

BACnetLiftGroupMode

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetLiftGroupMode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="unknown" />
    <xs:enumeration value="normal" />
    <xs:enumeration value="downPeak" />
    <xs:enumeration value="twoWay" />
    <xs:enumeration value="fourWay" />
    <xs:enumeration value="emergencyPower" />
    <xs:enumeration value="upPeak" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetLiftGroupMode>unknown</BACnetLiftGroupMode>
```

BACnetLightingCommand

This XML representation of complex data is used by the following objects/properties:

- **channel.presentValue**
- **timer.stateChangeValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetLightingCommand">
    <xs:sequence>
      <xs:element name="Operation" type="BACnetLightingOperation" />
      <xs:element minOccurs="0" name="TargetLevel" type="REAL" />
      <xs:element minOccurs="0" name="RampRate" type="REAL" />
      <xs:element minOccurs="0" name="StepIncrement" type="REAL" />
      <xs:element minOccurs="0" name="FadeTime" type="Unsigned" />
      <xs:element minOccurs="0" name="Priority" type="Unsigned" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetLightingCommand" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetLightingCommand>
    <Operation>none</Operation>
    <TargetLevel>1235.567749</TargetLevel>
    <RampRate>1236.567749</RampRate>
    <StepIncrement>1237.567749</StepIncrement>
    <FadeTime>1238</FadeTime>
    <Priority>1239</Priority>
  </BACnetLightingCommand>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetLightingCommand>
    <Operation>stepOff</Operation>
  </BACnetLightingCommand>
</Value>
```

BACnetLightingInProgress

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetLightingInProgress">
  <xs:restriction base="xs:string">
    <xs:enumeration value="idle" />
    <xs:enumeration value="fadeActive" />
    <xs:enumeration value="rampActive" />
    <xs:enumeration value="notControlled" />
    <xs:enumeration value="other" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetLightingInProgress>idle</BACnetLightingInProgress>
```

BACnetLightingOperation

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetLightingOperation">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none" />
    <xs:enumeration value="fadeTo" />
    <xs:enumeration value="rampTo" />
    <xs:enumeration value="stepUp" />
    <xs:enumeration value="stepDown" />
    <xs:enumeration value="stepOn" />
    <xs:enumeration value="stepOff" />
    <xs:enumeration value="warn" />
    <xs:enumeration value="warnOff" />
    <xs:enumeration value="warnRelinquish" />
    <xs:enumeration value="stop" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetLightingOperation>none</BACnetLightingOperation>
```

BACnetLightingTransition

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetLightingTransition">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none" />
    <xs:enumeration value="fade" />
    <xs:enumeration value="ramp" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetLightingTransition>none</BACnetLightingTransition>
```

BACnetLockStatus

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetLockStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="locked" />
    <xs:enumeration value="unlocked" />
    <xs:enumeration value="lockFault" />
    <xs:enumeration value="unused" />
    <xs:enumeration value="unknown" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetLockStatus>locked</BACnetLockStatus>
```

BACnetLogData

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetLogData">
  <xs:sequence>
    <xs:choice>
      <xs:element name="LogStatus" type="BACnetLogStatus" />
      <xs:element name="LogData" type="SequenceOfBACnetLogData_Unnamed" />
      <xs:element name="TimeChange" type="REAL" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetLogData>
  <LogStatus>
    <logDisabled>true</logDisabled>
    <logInterrupted>true</logInterrupted>
  </LogStatus>
</BACnetLogData>
```

Example 2:

```
<BACnetLogData>
  <LogData>
    <Unnamed>
      <BooleanValue>true</BooleanValue>
    </Unnamed>
  </LogData>
</BACnetLogData>
```

Example 3:

```
<BACnetLogData>
  <TimeChange>1234.567749</TimeChange>
</BACnetLogData>
```

BACnetLogData_Unnamed

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetLogData_Unnamed">
  <xs:sequence>
    <xs:choice>
      <xs:element name="BooleanValue" type="Boolean" />
      <xs:element name="RealValue" type="REAL" />
      <xs:element name="EnumeratedValue" type="Enumerated" />
      <xs:element name="UnsignedValue" type="Unsigned" />
      <xs:element name="IntegerValue" type="INTEGER" />
      <xs:element name="BitstringValue" type="BitString" />
      <xs:element name="NullValue" type="Null" />
      <xs:element name="Failure" type="Error" />
      <xs:element name="AnyValue" type="Any" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetLogData_Unnamed>
  <BooleanValue>true</BooleanValue>
</BACnetLogData_Unnamed>
```

Example 2:

```
<BACnetLogData_Unnamed>
  <RealValue>1234.567749</RealValue>
</BACnetLogData_Unnamed>
```

Example 3:

```
<BACnetLogData_Unnamed>
  <EnumeratedValue>1</EnumeratedValue>
</BACnetLogData_Unnamed>
```

Example 4:

```
<BACnetLogData_Unnamed>
  <UnsignedValue>1236</UnsignedValue>
</BACnetLogData_Unnamed>
```

Example 5:

```
<BACnetLogData_Unnamed>
  <IntegerValue>-797</IntegerValue>
</BACnetLogData_Unnamed>
```

Example 6:

```
<BACnetLogData_Unnamed>
  <BitstringValue>1</BitstringValue>
</BACnetLogData_Unnamed>
```

Example 7:

```
<BACnetLogData_Unnamed>
  <NullValue />
</BACnetLogData_Unnamed>
```

Example 8:

```
<BACnetLogData_Unnamed>
  <Failure>
    <ErrorClass>security</ErrorClass>
    <ErrorCode>inconsistentParameters</ErrorCode>
  </Failure>
</BACnetLogData_Unnamed>
```

Example 9:

```
<BACnetLogData_Unnamed>
  <AnyValue>
    <Boolean>true</Boolean>
  </AnyValue>
</BACnetLogData_Unnamed>
```

BACnetLogMultipleRecord

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetLogMultipleRecord">
  <xs:sequence>
    <xs:element name="Timestamp" type="BACnetDateTime" />
    <xs:element name="LogData" type="BACnetLogData" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetLogMultipleRecord>
  <Timestamp>
    <Date>2017-12-20</Date>
    <Time>22:39:15.980</Time>
  </Timestamp>
  <LogData>
    <LogStatus>
      <logDisabled>true</logDisabled>
      <logInterrupted>true</logInterrupted>
    </LogStatus>
  </LogData>
</BACnetLogMultipleRecord>
```

BACnetLogRecord

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetLogRecord">
  <xs:sequence>
    <xs:element name="Timestamp" type="BACnetDateTime" />
    <xs:element name="LogDatum" type="BACnetLogRecord_LogDatum" />
    <xs:element minOccurs="0" name="StatusFlags" type="BACnetStatusFlags" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetLogRecord>
  <Timestamp>
    <Date>2017-12-20</Date>
    <Time>22:39:15.980</Time>
  </Timestamp>
  <LogDatum>
    <LogStatus>
      <logDisabled>true</logDisabled>
      <logInterrupted>true</logInterrupted>
    </LogStatus>
  </LogDatum>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
</BACnetLogRecord>
```

Example 2:

```

<BACnetLogRecord>
  <Timestamp>
    <Date>2018-12-20</Date>
    <Time>22:39:15.980</Time>
  </Timestamp>
  <LogDatum>
    <LogStatus>
      <logDisabled>true</logDisabled>
      <logInterrupted>true</logInterrupted>
    </LogStatus>
  </LogDatum>
</BACnetLogRecord>

```

BACnetLogRecord_LogDatum

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetLogRecord_LogDatum">
  <xs:sequence>
    <xs:choice>
      <xs:element name="LogStatus" type="BACnetLogStatus" />
      <xs:element name="BooleanValue" type="Boolean" />
      <xs:element name="RealValue" type="REAL" />
      <xs:element name="EnumeratedValue" type="Enumerated" />
      <xs:element name="UnsignedValue" type="Unsigned" />
      <xs:element name="IntegerValue" type="INTEGER" />
      <xs:element name="BitstringValue" type="BitString" />
      <xs:element name="NullValue" type="Null" />
      <xs:element name="Failure" type="Error" />
      <xs:element name="TimeChange" type="REAL" />
      <xs:element name="AnyValue" type="Any" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```

<BACnetLogRecord_LogDatum>
  <LogStatus>
    <logDisabled>true</logDisabled>
    <logInterrupted>true</logInterrupted>
  </LogStatus>
</BACnetLogRecord_LogDatum>

```

Example 2:

```

<BACnetLogRecord_LogDatum>
  <BooleanValue>true</BooleanValue>
</BACnetLogRecord_LogDatum>

```

Example 3:

```

<BACnetLogRecord_LogDatum>
  <RealValue>1234.567749</RealValue>
</BACnetLogRecord_LogDatum>

```

Example 4:

```

<BACnetLogRecord_LogDatum>
  <EnumeratedValue>1</EnumeratedValue>
</BACnetLogRecord_LogDatum>

```

Example 5:

```

<BACnetLogRecord_LogDatum>
  <UnsignedValue>1236</UnsignedValue>
</BACnetLogRecord_LogDatum>

```

Example 6:


```
<BACnetLogRecord_LogDatum>
  <IntegerValue>-797</IntegerValue>
</BACnetLogRecord_LogDatum>
```

Example 7:

```
<BACnetLogRecord_LogDatum>
  <BitstringValue>1</BitstringValue>
</BACnetLogRecord_LogDatum>
```

Example 8:

```
<BACnetLogRecord_LogDatum>
  <NullValue />
</BACnetLogRecord_LogDatum>
```

Example 9:

```
<BACnetLogRecord_LogDatum>
  <Failure>
    <ErrorClass>security</ErrorClass>
    <ErrorCode>inconsistentParameters</ErrorCode>
  </Failure>
</BACnetLogRecord_LogDatum>
```

Example 10:

```
<BACnetLogRecord_LogDatum>
  <TimeChange>1240.567749</TimeChange>
</BACnetLogRecord_LogDatum>
```

Example 11:

```
<BACnetLogRecord_LogDatum>
  <AnyValue>
    <Boolean>true</Boolean>
  </AnyValue>
</BACnetLogRecord_LogDatum>
```

BACnetLogStatus

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetLogStatus">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="logDisabled" type="Boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="bufferPurged" type="Boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="logInterrupted" type="Boolean" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetLogStatus>
  <logDisabled>true</logDisabled>
  <logInterrupted>true</logInterrupted>
</BACnetLogStatus>
```

BACnetMaintenance

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:simpleType name="BACnetMaintenance">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none" />
    <xs:enumeration value="periodicTest" />
    <xs:enumeration value="needServiceOperational" />
    <xs:enumeration value="needServiceInoperative" />
  </xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<BACnetMaintenance>none</BACnetMaintenance>
```

BACnetNameValue

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetNameValue">
  <xs:sequence>
    <xs:element name="Name" type="CharacterString" />
    <xs:element minOccurs="0" name="Value" type="Any" />
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```

<BACnetNameValue>
  <Name>Abcde</Name>
  <Value>
    <Boolean>true</Boolean>
  </Value>
</BACnetNameValue>

```

Example 2:

```

<BACnetNameValue>
  <Name>Abcde</Name>
</BACnetNameValue>

```

BACnetNameValueCollection

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetNameValueCollection">
  <xs:sequence>
    <xs:element name="Members" type="SequenceOfBACnetNameValue" />
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```

<BACnetNameValueCollection>
  <Members />
</BACnetNameValueCollection>

```

Example 2:

```

<BACnetNameValueCollection>
  <Members>
    <BACnetNameValue>
      <Name>Abcde</Name>
      <Value>
        <Boolean>true</Boolean>
      </Value>
    </BACnetNameValue>
    <BACnetNameValue>
      <Name>Abcde</Name>
      <Value>
        <Boolean>true</Boolean>
      </Value>
    </BACnetNameValue>
    <BACnetNameValue>
      <Name>Abcde</Name>
      <Value>
        <Boolean>true</Boolean>
      </Value>
    </BACnetNameValue>
  </Members>
</BACnetNameValueCollection>

```

BACnetNetworkNumberQuality

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:simpleType name="BACnetNetworkNumberQuality">
  <xs:restriction base="xs:string">
    <xs:enumeration value="unknown" />
    <xs:enumeration value="learned" />
    <xs:enumeration value="learnedConfigured" />
    <xs:enumeration value="configured" />
  </xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<BACnetNetworkNumberQuality>unknown</BACnetNetworkNumberQuality>
```

BACnetNetworkPortCommand

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:simpleType name="BACnetNetworkPortCommand">
  <xs:restriction base="xs:string">
    <xs:enumeration value="idle" />
    <xs:enumeration value="discardChanges" />
    <xs:enumeration value="renewFdRegistration" />
    <xs:enumeration value="restartSlaveDiscovery" />
    <xs:enumeration value="renewDhcp" />
    <xs:enumeration value="restartAutonegotiation" />
    <xs:enumeration value="disconnect" />
    <xs:enumeration value="restartPort" />
  </xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<BACnetNetworkPortCommand>idle</BACnetNetworkPortCommand>
```

BACnetNetworkSecurityPolicy

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNetworkSecurityPolicy">
  <xs:sequence>
    <xs:element name="PortId" type="Unsigned8" />
    <xs:element name="SecurityLevel" type="BACnetSecurityPolicy" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNetworkSecurityPolicy>
  <PortId>33</PortId>
  <SecurityLevel>plainTrusted</SecurityLevel>
</BACnetNetworkSecurityPolicy>
```

BACnetNetworkType

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetNetworkType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ethernet" />
    <xs:enumeration value="arcnet" />
    <xs:enumeration value="mstp" />
    <xs:enumeration value="ptp" />
    <xs:enumeration value="lontalk" />
    <xs:enumeration value="ipv4" />
    <xs:enumeration value="zigbee" />
    <xs:enumeration value="virtual" />
    <xs:enumeration value="ipv6" />
    <xs:enumeration value="serial" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetNetworkType>ethernet</BACnetNetworkType>
```

BACnetNodeType

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:simpleType name="BACnetNodeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="unknown" />
    <xs:enumeration value="system" />
    <xs:enumeration value="network" />
    <xs:enumeration value="device" />
    <xs:enumeration value="organizational" />
    <xs:enumeration value="area" />
    <xs:enumeration value="equipment" />
    <xs:enumeration value="point" />
    <xs:enumeration value="collection" />
    <xs:enumeration value="property" />
    <xs:enumeration value="functional" />
    <xs:enumeration value="other" />
    <xs:enumeration value="subsystem" />
    <xs:enumeration value="building" />
    <xs:enumeration value="floor" />
    <xs:enumeration value="section" />
    <xs:enumeration value="module" />
    <xs:enumeration value="tree" />
    <xs:enumeration value="member" />
    <xs:enumeration value="protocol" />
    <xs:enumeration value="room" />
    <xs:enumeration value="zone" />
  </xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<BACnetNodeType>unknown</BACnetNodeType>
```

BACnetNotificationParameters

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetNotificationParameters">
  <xs:sequence>
    <xs:choice>
      <xs:element name="ChangeOfBitstring" type="BACnetNotificationParameters_ChangeOfBitstring" />
      <xs:element name="ChangeOfState" type="BACnetNotificationParameters_ChangeOfState" />
      <xs:element name="ChangeOfValue" type="BACnetNotificationParameters_ChangeOfValue" />
      <xs:element name="CommandFailure" type="BACnetNotificationParameters_CommandFailure" />
      <xs:element name="FloatingLimit" type="BACnetNotificationParameters_FloatingLimit" />
      <xs:element name="OutOfRange" type="BACnetNotificationParameters_OutOfRange" />
      <xs:element name="ComplexEventType" type="SequenceOfBACnetPropertyValue" />
      <xs:element name="ChangeOfLifeSafety"
type="BACnetNotificationParameters_ChangeOfLifeSafety" />
      <xs:element name="Extended" type="BACnetNotificationParameters_Extended" />
      <xs:element name="BufferReady" type="BACnetNotificationParameters_BufferReady" />
      <xs:element name="UnsignedRange" type="BACnetNotificationParameters_UnsignedRange" />
      <xs:element name="AccessEvent" type="BACnetNotificationParameters_AccessEvent" />
      <xs:element name="DoubleOutOfRange" type="BACnetNotificationParameters_DoubleOutOfRange" />
      <xs:element name="SignedOutOfRange" type="BACnetNotificationParameters_SignedOutOfRange" />
      <xs:element name="UnsignedOutOfRange"
type="BACnetNotificationParameters_UnsignedOutOfRange" />
      <xs:element name="ChangeOfCharacterstring"
type="BACnetNotificationParameters_ChangeOfCharacterstring" />
      <xs:element name="ChangeOfStatusFlags"
type="BACnetNotificationParameters_ChangeOfStatusFlags" />
      <xs:element name="ChangeOfReliability"
type="BACnetNotificationParameters_ChangeOfReliability" />
      <xs:element name="ChangeOfDiscreteValue"
type="BACnetNotificationParameters_ChangeOfDiscreteValue" />
      <xs:element name="ChangeOfTimer" type="BACnetNotificationParameters_ChangeOfTimer" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```

<BACnetNotificationParameters>
  <ChangeOfBitstring>
    <ReferencedBitstring>1</ReferencedBitstring>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
  </ChangeOfBitstring>
</BACnetNotificationParameters>

```

Example 2:

```

<BACnetNotificationParameters>
  <ChangeOfState>
    <NewState>
      <BooleanValue>true</BooleanValue>
    </NewState>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
  </ChangeOfState>
</BACnetNotificationParameters>

```

Example 3:

```

<BACnetNotificationParameters>
  <ChangeOfValue>
    <NewValue>
      <ChangedBits>1</ChangedBits>
    </NewValue>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
  </ChangeOfValue>
</BACnetNotificationParameters>

```

Example 4:

```

<BACnetNotificationParameters>
  <CommandFailure>
    <CommandValue>
      <Boolean>true</Boolean>
    </CommandValue>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
    <FeedbackValue>
      <Boolean>true</Boolean>
    </FeedbackValue>
  </CommandFailure>
</BACnetNotificationParameters>

```

Example 5:

```

<BACnetNotificationParameters>
  <FloatingLimit>
    <ReferenceValue>1234.567749</ReferenceValue>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
    <SetpointValue>1235.567749</SetpointValue>
    <ErrorLimit>1236.567749</ErrorLimit>
  </FloatingLimit>
</BACnetNotificationParameters>

```

Example 6:

```

<BACnetNotificationParameters>
  <OutOfRange>
    <ExceedingValue>1237.567749</ExceedingValue>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
    <Deadband>1238.567749</Deadband>
    <ExceededLimit>1239.567749</ExceededLimit>
  </OutOfRange>
</BACnetNotificationParameters>

```

Example 7:

```

<BACnetNotificationParameters>
  <ComplexEventType>
    <BACnetPropertyValue>
      <PropertyIdentifier>alarmValue</PropertyIdentifier>
      <PropertyArrayIndex>1241</PropertyArrayIndex>
      <PropertyValue>
        <Boolean>true</Boolean>
      </PropertyValue>
      <Priority>1242</Priority>
    </BACnetPropertyValue>
  </ComplexEventType>
</BACnetNotificationParameters>

```

Example 8:

```
<BACnetNotificationParameters>
  <ChangeOfLifeSafety>
    <NewState>testAlarm</NewState>
    <NewMode>disconnected</NewMode>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
    <OperationExpected>silence</OperationExpected>
  </ChangeOfLifeSafety>
</BACnetNotificationParameters>
```

Example 9:

```
<BACnetNotificationParameters>
  <Extended>
    <VendorId>7646</VendorId>
    <ExtendedEventType>1247</ExtendedEventType>
    <Parameters>
      <Unnamed>
        <Null />
      </Unnamed>
    </Parameters>
  </Extended>
</BACnetNotificationParameters>
```

Example 10:

```
<BACnetNotificationParameters>
  <BufferReady>
    <BufferProperty>
      <ObjectIdentifier>device-515</ObjectIdentifier>
      <PropertyIdentifier>changeOfStateCount</PropertyIdentifier>
      <PropertyArrayIndex>1250</PropertyArrayIndex>
      <DeviceIdentifier>device-518</DeviceIdentifier>
    </BufferProperty>
    <PreviousNotification>2363</PreviousNotification>
    <CurrentNotification>2364</CurrentNotification>
  </BufferReady>
</BACnetNotificationParameters>
```

Example 11:

```
<BACnetNotificationParameters>
  <UnsignedRange>
    <ExceedingValue>1254</ExceedingValue>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
    <ExceededLimit>1255</ExceededLimit>
  </UnsignedRange>
</BACnetNotificationParameters>
```

Example 12:


```
<BACnetNotificationParameters>
  <AccessEvent>
    <AccessEvent>deniedZoneNoAccessRights</AccessEvent>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
    <AccessEventTag>1257</AccessEventTag>
    <AccessEventTime>
      <Time>22:39:15.980</Time>
    </AccessEventTime>
    <AccessCredential>
      <DeviceIdentifier>device-526</DeviceIdentifier>
      <ObjectIdentifier>device-527</ObjectIdentifier>
    </AccessCredential>
    <AuthenticationFactor>
      <FormatType>custom</FormatType>
      <FormatClass>1262</FormatClass>
      <Value>FF</Value>
    </AuthenticationFactor>
  </AccessEvent>
</BACnetNotificationParameters>
```

Example 13:

```
<BACnetNotificationParameters>
  <DoubleOutOfRange>
    <ExceedingValue>123485.789012</ExceedingValue>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
    <Deadband>123486.789012</Deadband>
    <ExceededLimit>123487.789012</ExceededLimit>
  </DoubleOutOfRange>
</BACnetNotificationParameters>
```

Example 14:

```
<BACnetNotificationParameters>
  <SignedOutOfRange>
    <ExceedingValue>-768</ExceedingValue>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
    <Deadband>1267</Deadband>
    <ExceededLimit>-766</ExceededLimit>
  </SignedOutOfRange>
</BACnetNotificationParameters>
```

Example 15:

```
<BACnetNotificationParameters>
  <UnsignedOutOfRange>
    <ExceedingValue>1269</ExceedingValue>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
    <Deadband>1270</Deadband>
    <ExceededLimit>1271</ExceededLimit>
  </UnsignedOutOfRange>
</BACnetNotificationParameters>
```

Example 16:

```

<BACnetNotificationParameters>
  <ChangeOfCharacterstring>
    <ChangedValue>Abcde</ChangedValue>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
    <AlarmValue>Abcde</AlarmValue>
  </ChangeOfCharacterstring>
</BACnetNotificationParameters>

```

Example 17:

```

<BACnetNotificationParameters>
  <ChangeOfStatusFlags>
    <PresentValue>
      <Boolean>true</Boolean>
    </PresentValue>
    <ReferencedFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </ReferencedFlags>
  </ChangeOfStatusFlags>
</BACnetNotificationParameters>

```

Example 18:

```

<BACnetNotificationParameters>
  <ChangeOfReliability>
    <Reliability>tripped</Reliability>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
    <PropertyValues>
      <BACnetPropertyValue>
        <PropertyIdentifier>feedbackValue</PropertyIdentifier>
        <PropertyArrayIndex>1274</PropertyArrayIndex>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
        <Priority>1275</Priority>
      </BACnetPropertyValue>
    </PropertyValues>
  </ChangeOfReliability>
</BACnetNotificationParameters>

```

Example 19:

```

<BACnetNotificationParameters>
  <ChangeOfDiscreteValue>
    <NewValue>
      <Boolean>true</Boolean>
    </NewValue>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
  </ChangeOfDiscreteValue>
</BACnetNotificationParameters>

```

Example 20:

```

<BACnetNotificationParameters>
  <ChangeOfTimer>
    <NewState>idle</NewState>
    <StatusFlags>
      <inAlarm>true</inAlarm>
      <overridden>true</overridden>
    </StatusFlags>
    <UpdateTime>
      <Date>2060-12-20</Date>
      <Time>22:39:15.980</Time>
    </UpdateTime>
    <LastStateChange>runningToExpired</LastStateChange>
    <InitialTimeout>1279</InitialTimeout>
    <ExpirationTime>
      <Date>2063-12-20</Date>
      <Time>22:39:15.980</Time>
    </ExpirationTime>
  </ChangeOfTimer>
</BACnetNotificationParameters>

```

BACnetNotificationParameters_AccessEvent

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetNotificationParameters_AccessEvent">
  <xs:sequence>
    <xs:element name="AccessEvent" type="BACnetAccessEvent" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
    <xs:element name="AccessEventTag" type="Unsigned" />
    <xs:element name="AccessEventTime" type="BACnetTimeStamp" />
    <xs:element name="AccessCredential" type="BACnetDeviceObjectReference" />
    <xs:element minOccurs="0" name="AuthenticationFactor" type="BACnetAuthenticationFactor" />
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```

<BACnetNotificationParameters_AccessEvent>
  <AccessEvent>none</AccessEvent>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <AccessEventTag>1235</AccessEventTag>
  <AccessEventTime>
    <Time>22:39:15.980</Time>
  </AccessEventTime>
  <AccessCredential>
    <DeviceIdentifier>device-504</DeviceIdentifier>
    <ObjectIdentifier>device-505</ObjectIdentifier>
  </AccessCredential>
  <AuthenticationFactor>
    <FormatType>simpleNumber56</FormatType>
    <FormatClass>1240</FormatClass>
    <Value>00010203040506070809</Value>
  </AuthenticationFactor>
</BACnetNotificationParameters_AccessEvent>

```

Example 2:

```
<BACnetNotificationParameters_AccessEvent>
  <AccessEvent>lockoutOther</AccessEvent>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <AccessEventTag>1242</AccessEventTag>
  <AccessEventTime>
    <Time>22:39:15.980</Time>
  </AccessEventTime>
  <AccessCredential>
    <DeviceIdentifier>device-511</DeviceIdentifier>
    <ObjectIdentifier>device-512</ObjectIdentifier>
  </AccessCredential>
</BACnetNotificationParameters_AccessEvent>
```

BACnetNotificationParameters_BufferReady

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_BufferReady">
  <xs:sequence>
    <xs:element name="BufferProperty" type="BACnetDeviceObjectPropertyReference" />
    <xs:element name="PreviousNotification" type="Unsigned32" />
    <xs:element name="CurrentNotification" type="Unsigned32" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_BufferReady>
  <BufferProperty>
    <ObjectIdentifier>device-501</ObjectIdentifier>
    <PropertyIdentifier>ackRequired</PropertyIdentifier>
    <PropertyArrayIndex>1236</PropertyArrayIndex>
    <DeviceIdentifier>device-504</DeviceIdentifier>
  </BufferProperty>
  <PreviousNotification>2349</PreviousNotification>
  <CurrentNotification>2350</CurrentNotification>
</BACnetNotificationParameters_BufferReady>
```

BACnetNotificationParameters_ChangeOfBitstring

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_ChangeOfBitstring">
  <xs:sequence>
    <xs:element name="ReferencedBitstring" type="BitString" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_ChangeOfBitstring>
  <ReferencedBitstring>101</ReferencedBitstring>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
</BACnetNotificationParameters_ChangeOfBitstring>
```

BACnetNotificationParameters_ChangeOfCharacterstring

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_ChangeOfCharacterstring">
  <xs:sequence>
    <xs:element name="ChangedValue" type="CharacterString" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
    <xs:element name="AlarmValue" type="CharacterString" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_ChangeOfCharacterstring>
  <ChangedValue>Abcde</ChangedValue>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <AlarmValue>Abcde</AlarmValue>
</BACnetNotificationParameters_ChangeOfCharacterstring>
```

BACnetNotificationParameters_ChangeOfDiscreteValue

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_ChangeOfDiscreteValue">
  <xs:sequence>
    <xs:element name="NewValue"
type="BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_ChangeOfDiscreteValue>
  <NewValue>
    <Boolean>true</Boolean>
  </NewValue>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
</BACnetNotificationParameters_ChangeOfDiscreteValue>
```

BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Boolean" type="Boolean" />
      <xs:element name="Unsigned" type="Unsigned" />
      <xs:element name="Integer" type="INTEGER" />
      <xs:element name="Enumerated" type="Enumerated" />
      <xs:element name="Characterstring" type="CharacterString" />
      <xs:element name="Octetstring" type="OctetString" />
      <xs:element name="Date" type="Date" />
      <xs:element name="Time" type="Time" />
      <xs:element name="Objectidentifier" type="BACnetObjectIdentifier" />
      <xs:element name="Datetime" type="BACnetDateTime" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
  <Boolean>true</Boolean>
</BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
```

Example 2:

```
<BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
  <Unsigned>1234</Unsigned>
</BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
```

Example 3:

```
<BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
  <Integer>-799</Integer>
</BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
```

Example 4:

```
<BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
  <Enumerated>2</Enumerated>
</BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
```

Example 5:

```
<BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
  <Characterstring>Abcde</Characterstring>
</BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
```

Example 6:

```
<BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
  <Octetstring>FF</Octetstring>
</BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
```

Example 7:

```
<BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
  <Date>2020-12-20</Date>
</BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
```

Example 8:

```
<BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
  <Time>22:39:15.980</Time>
</BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
```

Example 9:

```
<BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
  <Objectidentifier>device-506</Objectidentifier>
</BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
```

Example 10:

```
<BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
  <Datetime>
    <Date>2023-12-20</Date>
    <Time>22:39:15.980</Time>
  </Datetime>
</BACnetNotificationParameters_ChangeOfDiscreteValue_NewValue>
```

BACnetNotificationParameters_ChangeOfLifeSafety

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_ChangeOfLifeSafety">
  <xs:sequence>
    <xs:element name="NewState" type="BACnetLifeSafetyState" />
    <xs:element name="NewMode" type="BACnetLifeSafetyMode" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
    <xs:element name="OperationExpected" type="BACnetLifeSafetyOperation" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_ChangeOfLifeSafety>
  <NewState>quiet</NewState>
  <NewMode>on</NewMode>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <OperationExpected>silenceAudible</OperationExpected>
</BACnetNotificationParameters_ChangeOfLifeSafety>
```

BACnetNotificationParameters_ChangeOfReliability

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_ChangeOfReliability">
  <xs:sequence>
    <xs:element name="Reliability" type="BACnetReliability" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
    <xs:element name="PropertyValues" type="SequenceOfBACnetPropertyValue" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_ChangeOfReliability>
  <Reliability>noFaultDetected</Reliability>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <PropertyValues />
</BACnetNotificationParameters_ChangeOfReliability>
```

Example 2:

```
<BACnetNotificationParameters_ChangeOfReliability>
  <Reliability>noSensor</Reliability>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <PropertyValues>
    <BACnetPropertyValue>
      <PropertyIdentifier>action</PropertyIdentifier>
      <PropertyArrayIndex>1237</PropertyArrayIndex>
      <PropertyValue>
        <Boolean>true</Boolean>
      </PropertyValue>
      <Priority>1238</Priority>
    </BACnetPropertyValue>
    <BACnetPropertyValue>
      <PropertyIdentifier>activeVtSessions</PropertyIdentifier>
      <PropertyArrayIndex>1240</PropertyArrayIndex>
      <PropertyValue>
        <Boolean>true</Boolean>
      </PropertyValue>
      <Priority>1241</Priority>
    </BACnetPropertyValue>
    <BACnetPropertyValue>
      <PropertyIdentifier>all</PropertyIdentifier>
      <PropertyArrayIndex>1243</PropertyArrayIndex>
      <PropertyValue>
        <Boolean>true</Boolean>
      </PropertyValue>
      <Priority>1244</Priority>
    </BACnetPropertyValue>
  </PropertyValues>
</BACnetNotificationParameters_ChangeOfReliability>
```

BACnetNotificationParameters_ChangeOfState

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_ChangeOfState">
  <xs:sequence>
    <xs:element name="NewState" type="BACnetPropertyStates" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_ChangeOfState>
  <NewState>
    <BooleanValue>true</BooleanValue>
  </NewState>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
</BACnetNotificationParameters_ChangeOfState>
```

BACnetNotificationParameters_ChangeOfStatusFlags

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema


```
<xs:complexType name="BACnetNotificationParameters_ChangeOfStatusFlags">
  <xs:sequence>
    <xs:element minOccurs="0" name="PresentValue" type="Any" />
    <xs:element name="ReferencedFlags" type="BACnetStatusFlags" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_ChangeOfStatusFlags>
  <PresentValue>
    <Boolean>true</Boolean>
  </PresentValue>
  <ReferencedFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </ReferencedFlags>
</BACnetNotificationParameters_ChangeOfStatusFlags>
```

Example 2:

```
<BACnetNotificationParameters_ChangeOfStatusFlags>
  <ReferencedFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </ReferencedFlags>
</BACnetNotificationParameters_ChangeOfStatusFlags>
```

BACnetNotificationParameters_ChangeOfTimer

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_ChangeOfTimer">
  <xs:sequence>
    <xs:element name="NewState" type="BACnetTimerState" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
    <xs:element name="UpdateTime" type="BACnetDateTime" />
    <xs:element minOccurs="0" name="LastStateChange" type="BACnetTimerTransition" />
    <xs:element minOccurs="0" name="InitialTimeout" type="Unsigned" />
    <xs:element minOccurs="0" name="ExpirationTime" type="BACnetDateTime" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_ChangeOfTimer>
  <NewState>idle</NewState>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <UpdateTime>
    <Date>2018-12-20</Date>
    <Time>22:39:15.980</Time>
  </UpdateTime>
  <LastStateChange>runningToIdle</LastStateChange>
  <InitialTimeout>1237</InitialTimeout>
  <ExpirationTime>
    <Date>2021-12-20</Date>
    <Time>22:39:15.980</Time>
  </ExpirationTime>
</BACnetNotificationParameters_ChangeOfTimer>
```

Example 2:

```
<BACnetNotificationParameters_ChangeOfTimer>
  <NewState>expired</NewState>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <UpdateTime>
    <Date>2023-12-20</Date>
    <Time>22:39:15.980</Time>
  </UpdateTime>
</BACnetNotificationParameters_ChangeOfTimer>
```

BACnetNotificationParameters_ChangeOfValue

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_ChangeOfValue">
  <xs:sequence>
    <xs:element name="NewValue" type="BACnetNotificationParameters_ChangeOfValue_NewValue" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_ChangeOfValue>
  <NewValue>
    <ChangedBits>1</ChangedBits>
  </NewValue>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
</BACnetNotificationParameters_ChangeOfValue>
```

BACnetNotificationParameters_ChangeOfValue_NewValue

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_ChangeOfValue_NewValue">
  <xs:sequence>
    <xs:choice>
      <xs:element name="ChangedBits" type="BitString" />
      <xs:element name="ChangedValue" type="REAL" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_ChangeOfValue_NewValue>
  <ChangedBits>1</ChangedBits>
</BACnetNotificationParameters_ChangeOfValue_NewValue>
```

Example 2:

```
<BACnetNotificationParameters_ChangeOfValue_NewValue>
  <ChangedValue>1234.567749</ChangedValue>
</BACnetNotificationParameters_ChangeOfValue_NewValue>
```

BACnetNotificationParameters_CommandFailure

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_CommandFailure">
  <xs:sequence>
    <xs:element name="CommandValue" type="Any" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
    <xs:element name="FeedbackValue" type="Any" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_CommandFailure>
  <CommandValue>
    <Boolean>true</Boolean>
  </CommandValue>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <FeedbackValue>
    <Boolean>true</Boolean>
  </FeedbackValue>
</BACnetNotificationParameters_CommandFailure>
```

BACnetNotificationParameters_DoubleOutOfRange

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_DoubleOutOfRange">
  <xs:sequence>
    <xs:element name="ExceedingValue" type="Double" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
    <xs:element name="Deadband" type="Double" />
    <xs:element name="ExceededLimit" type="Double" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_DoubleOutOfRange>
  <ExceedingValue>123456.789012</ExceedingValue>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <Deadband>123457.789012</Deadband>
  <ExceededLimit>123458.789012</ExceededLimit>
</BACnetNotificationParameters_DoubleOutOfRange>
```

BACnetNotificationParameters_Extended

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_Extended">
  <xs:sequence>
    <xs:element name="VendorId" type="Unsigned16" />
    <xs:element name="ExtendedEventType" type="Unsigned" />
    <xs:element name="Parameters" type="SequenceOfBACnetNotificationParameters_Extended_Unnamed" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_Extended>
  <VendorId>7634</VendorId>
  <ExtendedEventType>1235</ExtendedEventType>
  <Parameters />
</BACnetNotificationParameters_Extended>
```

Example 2:

```
<BACnetNotificationParameters_Extended>
  <VendorId>7636</VendorId>
  <ExtendedEventType>1237</ExtendedEventType>
  <Parameters>
    <Unnamed>
      <Null />
    </Unnamed>
    <Unnamed>
      <Null />
    </Unnamed>
    <Unnamed>
      <Null />
    </Unnamed>
    <Unnamed>
      <Null />
    </Unnamed>
  </Parameters>
</BACnetNotificationParameters_Extended>
```

BACnetNotificationParameters_Extended_Unnamed

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_Extended_Unnamed">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Null" type="Null" />
      <xs:element name="Real" type="REAL" />
      <xs:element name="Unsigned" type="Unsigned" />
      <xs:element name="Boolean" type="Boolean" />
      <xs:element name="Integer" type="INTEGER" />
      <xs:element name="Double" type="Double" />
      <xs:element name="Octetstring" type="OctetString" />
      <xs:element name="Characterstring" type="CharacterString" />
      <xs:element name="Bitstring" type="BitString" />
      <xs:element name="Enumerated" type="Enumerated" />
      <xs:element name="Date" type="Date" />
      <xs:element name="Time" type="Time" />
      <xs:element name="Objectidentifier" type="BACnetObjectIdentifier" />
      <xs:element name="PropertyValue" type="BACnetDeviceObjectPropertyValue" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <Null />
</BACnetNotificationParameters_Extended_Unnamed>
```

Example 2:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <Real>1234.567749</Real>
</BACnetNotificationParameters_Extended_Unnamed>
```

Example 3:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <Unsigned>1235</Unsigned>
</BACnetNotificationParameters_Extended_Unnamed>
```

Example 4:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <Boolean>true</Boolean>
</BACnetNotificationParameters_Extended_Unnamed>
```

Example 5:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <Integer>-798</Integer>
</BACnetNotificationParameters_Extended_Unnamed>
```

Example 6:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <Double>123459.789012</Double>
</BACnetNotificationParameters_Extended_Unnamed>
```

Example 7:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <Octetstring>FF</Octetstring>
</BACnetNotificationParameters_Extended_Unnamed>
```

Example 8:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <Characterstring>Abcde</Characterstring>
</BACnetNotificationParameters_Extended_Unnamed>
```

Example 9:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <Bitstring>1</Bitstring>
</BACnetNotificationParameters_Extended_Unnamed>
```

Example 10:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <Enumerated>4</Enumerated>
</BACnetNotificationParameters_Extended_Unnamed>
```

Example 11:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <Date>2022-12-20</Date>
</BACnetNotificationParameters_Extended_Unnamed>
```

Example 12:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <Time>22:39:15.980</Time>
</BACnetNotificationParameters_Extended_Unnamed>
```

Example 13:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <Objectidentifier>device-508</Objectidentifier>
</BACnetNotificationParameters_Extended_Unnamed>
```

Example 14:

```
<BACnetNotificationParameters_Extended_Unnamed>
  <PropertyValue>
    <DeviceIdentifier>device-509</DeviceIdentifier>
    <ObjectIdentifier>device-510</ObjectIdentifier>
    <PropertyIdentifier>apduSegmentTimeout</PropertyIdentifier>
    <PropertyArrayIndex>1245</PropertyArrayIndex>
    <PropertyValue>
      <Unsigned>1246</Unsigned>
    </PropertyValue>
  </PropertyValue>
</BACnetNotificationParameters_Extended_Unnamed>
```

BACnetNotificationParameters_FloatingLimit

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_FloatingLimit">
  <xs:sequence>
    <xs:element name="ReferenceValue" type="REAL" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
    <xs:element name="SetpointValue" type="REAL" />
    <xs:element name="ErrorLimit" type="REAL" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_FloatingLimit>
  <ReferenceValue>1234.567749</ReferenceValue>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <SetpointValue>1235.567749</SetpointValue>
  <ErrorLimit>1236.567749</ErrorLimit>
</BACnetNotificationParameters_FloatingLimit>
```

BACnetNotificationParameters_OutOfRange

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_OutOfRange">
  <xs:sequence>
    <xs:element name="ExceedingValue" type="REAL" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
    <xs:element name="Deadband" type="REAL" />
    <xs:element name="ExceededLimit" type="REAL" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_OutOfRange>
  <ExceedingValue>1234.567749</ExceedingValue>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <Deadband>1235.567749</Deadband>
  <ExceededLimit>1236.567749</ExceededLimit>
</BACnetNotificationParameters_OutOfRange>
```

BACnetNotificationParameters_SignedOutOfRange

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_SignedOutOfRange">
  <xs:sequence>
    <xs:element name="ExceedingValue" type="INTEGER" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
    <xs:element name="Deadband" type="Unsigned" />
    <xs:element name="ExceededLimit" type="INTEGER" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_SignedOutOfRange>
  <ExceedingValue>-800</ExceedingValue>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <Deadband>1235</Deadband>
  <ExceededLimit>-798</ExceededLimit>
</BACnetNotificationParameters_SignedOutOfRange>
```

BACnetNotificationParameters_UnsignedOutOfRange

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_UnsignedOutOfRange">
  <xs:sequence>
    <xs:element name="ExceedingValue" type="Unsigned" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
    <xs:element name="Deadband" type="Unsigned" />
    <xs:element name="ExceededLimit" type="Unsigned" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_UnsignedOutOfRange>
  <ExceedingValue>1234</ExceedingValue>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <Deadband>1235</Deadband>
  <ExceededLimit>1236</ExceededLimit>
</BACnetNotificationParameters_UnsignedOutOfRange>
```

BACnetNotificationParameters_UnsignedRange

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetNotificationParameters_UnsignedRange">
  <xs:sequence>
    <xs:element name="ExceedingValue" type="Unsigned" />
    <xs:element name="StatusFlags" type="BACnetStatusFlags" />
    <xs:element name="ExceededLimit" type="Unsigned" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetNotificationParameters_UnsignedRange>
  <ExceedingValue>1234</ExceedingValue>
  <StatusFlags>
    <inAlarm>true</inAlarm>
    <overridden>true</overridden>
  </StatusFlags>
  <ExceededLimit>1235</ExceededLimit>
</BACnetNotificationParameters_UnsignedRange>
```

BACnetNotifyType

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetNotifyType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="alarm" />
    <xs:enumeration value="event" />
    <xs:enumeration value="ackNotification" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetNotifyType>alarm</BACnetNotifyType>
```

BACnetObjectIdentifier

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.priorityArray**
- **analogOutput.priorityArray**
- **analogValue.priorityArray**
- **binaryLightingOutput.priorityArray**
- **binaryOutput.priorityArray**
- **binaryValue.priorityArray**
- **bitstringValue.priorityArray**
- **channel.presentValue**
- **characterstringValue.priorityArray**
- **dateValue.priorityArray**
- **datepatternValue.priorityArray**
- **datetimeValue.priorityArray**

- **datetimepatternValue.priorityArray**
- **integerValue.priorityArray**
- **largeAnalogValue.priorityArray**
- **lightingOutput.priorityArray**
- **multiStateOutput.priorityArray**
- **multiStateValue.priorityArray**
- **octetstringValue.priorityArray**
- **positiveIntegerValue.priorityArray**
- **timeValue.priorityArray**
- **timepatternValue.priorityArray**
- **timer.stateChangeValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="BACnetObjectIdentifier">
    <xs:restriction base="xs:string" />
  </xs:simpleType>
  <xs:element name="Value" type="BACnetObjectIdentifier" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetObjectIdentifier>device-501</BACnetObjectIdentifier>
</Value>
```

BACnetObjectPropertyReference

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetObjectPropertyReference">
  <xs:sequence>
    <xs:element name="ObjectIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="PropertyIdentifier" type="BACnetPropertyIdentifier" />
    <xs:element minOccurs="0" name="PropertyArrayIndex" type="Unsigned" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetObjectPropertyReference>
  <ObjectIdentifier>device-501</ObjectIdentifier>
  <PropertyIdentifier>ackRequired</PropertyIdentifier>
  <PropertyArrayIndex>1236</PropertyArrayIndex>
</BACnetObjectPropertyReference>
```

Example 2:

```
<BACnetObjectPropertyReference>
  <ObjectIdentifier>device-504</ObjectIdentifier>
  <PropertyIdentifier>activeText</PropertyIdentifier>
</BACnetObjectPropertyReference>
```

BACnetOptionalCharacterString

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetOptionalCharacterString">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Null" type="Null" />
      <xs:element name="Characterstring" type="CharacterString" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetOptionalCharacterString>
  <Null />
</BACnetOptionalCharacterString>
```

Example 2:

```
<BACnetOptionalCharacterString>
  <Characterstring>Abcde</Characterstring>
</BACnetOptionalCharacterString>
```

BACnetPolarity

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetPolarity">
  <xs:restriction base="xs:string">
    <xs:enumeration value="normal" />
    <xs:enumeration value="reverse" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetPolarity>normal</BACnetPolarity>
```

BACnetPortPermission

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetPortPermission">
  <xs:sequence>
    <xs:element name="PortId" type="Unsigned8" />
    <xs:element name="Enabled" type="Boolean" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetPortPermission>
  <PortId>33</PortId>
  <Enabled>true</Enabled>
</BACnetPortPermission>
```

BACnetProgramError

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetProgramError">
  <xs:restriction base="xs:string">
    <xs:enumeration value="normal" />
    <xs:enumeration value="loadFailed" />
    <xs:enumeration value="internal" />
    <xs:enumeration value="program" />
    <xs:enumeration value="other" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetProgramError>normal</BACnetProgramError>
```

BACnetProgramRequest

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetProgramRequest">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ready" />
    <xs:enumeration value="load" />
    <xs:enumeration value="run" />
    <xs:enumeration value="halt" />
    <xs:enumeration value="restart" />
    <xs:enumeration value="unload" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetProgramRequest>ready</BACnetProgramRequest>
```

BACnetProgramState

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetProgramState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="idle" />
    <xs:enumeration value="loading" />
    <xs:enumeration value="running" />
    <xs:enumeration value="waiting" />
    <xs:enumeration value="halted" />
    <xs:enumeration value="unloading" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetProgramState>idle</BACnetProgramState>
```

BACnetPropertyAccessResult

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetPropertyAccessResult">
  <xs:sequence>
    <xs:element name="ObjectIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="PropertyIdentifier" type="BACnetPropertyIdentifier" />
    <xs:element minOccurs="0" name="PropertyArrayIndex" type="Unsigned" />
    <xs:element minOccurs="0" name="DeviceIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="AccessResult" type="BACnetPropertyAccessResult_AccessResult" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetPropertyAccessResult>
  <ObjectIdentifier>device-501</ObjectIdentifier>
  <PropertyIdentifier>ackRequired</PropertyIdentifier>
  <PropertyArrayIndex>1236</PropertyArrayIndex>
  <DeviceIdentifier>device-504</DeviceIdentifier>
  <AccessResult>
    <PropertyValue>
      <Boolean>true</Boolean>
    </PropertyValue>
  </AccessResult>
</BACnetPropertyAccessResult>
```

Example 2:

```
<BACnetPropertyAccessResult>
  <ObjectIdentifier>device-505</ObjectIdentifier>
  <PropertyIdentifier>activeVtSessions</PropertyIdentifier>
  <AccessResult>
    <PropertyValue>
      <SequenceOfBACnetVTSession>
        <BACnetVTSession>
          <LocalVtSessionId>39</LocalVtSessionId>
          <RemoteVtSessionId>40</RemoteVtSessionId>
          <RemoteVtAddress>
            <NetworkNumber>7642</NetworkNumber>
            <MacAddress>FF</MacAddress>
          </RemoteVtAddress>
        </BACnetVTSession>
      </SequenceOfBACnetVTSession>
    </PropertyValue>
  </AccessResult>
</BACnetPropertyAccessResult>
```

BACnetPropertyAccessResult_AccessResult

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetPropertyAccessResult_AccessResult">
  <xs:sequence>
    <xs:choice>
      <xs:element name="PropertyValue" type="Any" />
      <xs:element name="PropertyAccessError" type="Error" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetPropertyAccessResult_AccessResult>
  <PropertyValue>
    <Boolean>true</Boolean>
  </PropertyValue>
</BACnetPropertyAccessResult_AccessResult>
```

Example 2:

```
<BACnetPropertyAccessResult_AccessResult>
  <PropertyAccessError>
    <ErrorClass>device</ErrorClass>
    <ErrorCode>configurationInProgress</ErrorCode>
  </PropertyAccessError>
</BACnetPropertyAccessResult_AccessResult>
```

BACnetPropertyIdentifier

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:simpleType name="BACnetPropertyIdentifier">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ackedTransitions" />
    <xs:enumeration value="ackRequired" />
    <xs:enumeration value="action" />
    <xs:enumeration value="actionText" />
    <xs:enumeration value="activeText" />
    <xs:enumeration value="activeVtSessions" />
    <xs:enumeration value="alarmValue" />
    <xs:enumeration value="alarmValues" />
    <xs:enumeration value="all" />
    <xs:enumeration value="allWritesSuccessful" />
    <xs:enumeration value="apduSegmentTimeout" />
    <xs:enumeration value="apduTimeout" />
    <xs:enumeration value="applicationSoftwareVersion" />
    <xs:enumeration value="archive" />
    <xs:enumeration value="bias" />
    <xs:enumeration value="changeOfStateCount" />
    <xs:enumeration value="changeOfStateTime" />
    <xs:enumeration value="notificationClass" />
    <xs:enumeration value="controlledVariableReference" />
    <xs:enumeration value="controlledVariableUnits" />
    <xs:enumeration value="controlledVariableValue" />
    <xs:enumeration value="covIncrement" />
    <xs:enumeration value="dateList" />
    <xs:enumeration value="daylightSavingsStatus" />
    <xs:enumeration value="deadband" />
    <xs:enumeration value="derivativeConstant" />
    <xs:enumeration value="derivativeConstantUnits" />
    <xs:enumeration value="description" />
    <xs:enumeration value="descriptionOfHalt" />
    <xs:enumeration value="deviceAddressBinding" />
    <xs:enumeration value="deviceType" />
    <xs:enumeration value="effectivePeriod" />
    <xs:enumeration value="elapsedActiveTime" />
    <xs:enumeration value="errorLimit" />
    <xs:enumeration value="eventEnable" />
    <xs:enumeration value="eventState" />
    <xs:enumeration value="eventType" />
    <xs:enumeration value="exceptionSchedule" />
    <xs:enumeration value="faultValues" />
    <xs:enumeration value="feedbackValue" />
    <xs:enumeration value="fileAccessMethod" />
    <xs:enumeration value="fileSize" />
    <xs:enumeration value="fileType" />
    <xs:enumeration value="firmwareRevision" />
    <xs:enumeration value="highLimit" />
    <xs:enumeration value="inactiveText" />
    <xs:enumeration value="inProcess" />
    <xs:enumeration value="instanceOf" />
    <xs:enumeration value="integralConstant" />
    <xs:enumeration value="integralConstantUnits" />
    <xs:enumeration value="limitEnable" />
    <xs:enumeration value="listOfGroupMembers" />
    <xs:enumeration value="listOfObjectPropertyReferences" />
    <xs:enumeration value="localDate" />
    <xs:enumeration value="localTime" />
    <xs:enumeration value="location" />
    <xs:enumeration value="lowLimit" />
    <xs:enumeration value="manipulatedVariableReference" />
    <xs:enumeration value="maximumOutput" />
    <xs:enumeration value="maxApduLengthAccepted" />
    <xs:enumeration value="maxInfoFrames" />
    <xs:enumeration value="maxMaster" />
    <xs:enumeration value="maxPresValue" />
    <xs:enumeration value="minimumOffTime" />
    <xs:enumeration value="minimumOnTime" />
    <xs:enumeration value="minimumOutput" />
    <xs:enumeration value="minPresValue" />
    <xs:enumeration value="modelName" />
    <xs:enumeration value="modificationDate" />
  
```

```

<xs:enumeration value="notifyType" />
<xs:enumeration value="numberOfAduRetries" />
<xs:enumeration value="numberOfStates" />
<xs:enumeration value="objectIdentifier" />
<xs:enumeration value="objectList" />
<xs:enumeration value="objectName" />
<xs:enumeration value="objectPropertyReference" />
<xs:enumeration value="objectType" />
<xs:enumeration value="optional" />
<xs:enumeration value="outOfService" />
<xs:enumeration value="outputUnits" />
<xs:enumeration value="eventParameters" />
<xs:enumeration value="polarity" />
<xs:enumeration value="presentValue" />
<xs:enumeration value="priority" />
<xs:enumeration value="priorityArray" />
<xs:enumeration value="priorityForWriting" />
<xs:enumeration value="processIdentifier" />
<xs:enumeration value="programChange" />
<xs:enumeration value="programLocation" />
<xs:enumeration value="programState" />
<xs:enumeration value="proportionalConstant" />
<xs:enumeration value="proportionalConstantUnits" />
<xs:enumeration value="protocolObjectTypesSupported" />
<xs:enumeration value="protocolServicesSupported" />
<xs:enumeration value="protocolVersion" />
<xs:enumeration value="readOnly" />
<xs:enumeration value="reasonForHalt" />
<xs:enumeration value="recipientList" />
<xs:enumeration value="reliability" />
<xs:enumeration value="relinquishDefault" />
<xs:enumeration value="required" />
<xs:enumeration value="resolution" />
<xs:enumeration value="segmentationSupported" />
<xs:enumeration value="setpoint" />
<xs:enumeration value="setpointReference" />
<xs:enumeration value="stateText" />
<xs:enumeration value="statusFlags" />
<xs:enumeration value="systemStatus" />
<xs:enumeration value="timeDelay" />
<xs:enumeration value="timeOfActiveTimeReset" />
<xs:enumeration value="timeOfStateCountReset" />
<xs:enumeration value="timeSynchronizationRecipients" />
<xs:enumeration value="units" />
<xs:enumeration value="updateInterval" />
<xs:enumeration value="utcOffset" />
<xs:enumeration value="vendorIdentifier" />
<xs:enumeration value="vendorName" />
<xs:enumeration value="vtClassesSupported" />
<xs:enumeration value="weeklySchedule" />
<xs:enumeration value="attemptedSamples" />
<xs:enumeration value="averageValue" />
<xs:enumeration value="bufferSize" />
<xs:enumeration value="clientCovIncrement" />
<xs:enumeration value="covResubscriptionInterval" />
<xs:enumeration value="eventTimeStamps" />
<xs:enumeration value="logBuffer" />
<xs:enumeration value="logDeviceObjectProperty" />
<xs:enumeration value="enable" />
<xs:enumeration value="logInterval" />
<xs:enumeration value="maximumValue" />
<xs:enumeration value="minimumValue" />
<xs:enumeration value="notificationThreshold" />
<xs:enumeration value="protocolRevision" />
<xs:enumeration value="recordsSinceNotification" />
<xs:enumeration value="recordCount" />
<xs:enumeration value="startTime" />
<xs:enumeration value="stopTime" />
<xs:enumeration value="stopWhenFull" />
<xs:enumeration value="totalRecordCount" />
<xs:enumeration value="validSamples" />

```

```

<xs:enumeration value="windowInterval" />
<xs:enumeration value="windowSamples" />
<xs:enumeration value="maximumValueTimestamp" />
<xs:enumeration value="minimumValueTimestamp" />
<xs:enumeration value="varianceValue" />
<xs:enumeration value="activeCovSubscriptions" />
<xs:enumeration value="backupFailureTimeout" />
<xs:enumeration value="configurationFiles" />
<xs:enumeration value="databaseRevision" />
<xs:enumeration value="directReading" />
<xs:enumeration value="lastRestoreTime" />
<xs:enumeration value="maintenanceRequired" />
<xs:enumeration value="memberOf" />
<xs:enumeration value="mode" />
<xs:enumeration value="operationExpected" />
<xs:enumeration value="setting" />
<xs:enumeration value="silenced" />
<xs:enumeration value="trackingValue" />
<xs:enumeration value="zoneMembers" />
<xs:enumeration value="lifeSafetyAlarmValues" />
<xs:enumeration value="maxSegmentsAccepted" />
<xs:enumeration value="profileName" />
<xs:enumeration value="autoSlaveDiscovery" />
<xs:enumeration value="manualSlaveAddressBinding" />
<xs:enumeration value="slaveAddressBinding" />
<xs:enumeration value="slaveProxyEnable" />
<xs:enumeration value="lastNotifyRecord" />
<xs:enumeration value="scheduleDefault" />
<xs:enumeration value="acceptedModes" />
<xs:enumeration value="adjustValue" />
<xs:enumeration value="count" />
<xs:enumeration value="countBeforeChange" />
<xs:enumeration value="countChangeTime" />
<xs:enumeration value="covPeriod" />
<xs:enumeration value="inputReference" />
<xs:enumeration value="limitMonitoringInterval" />
<xs:enumeration value="loggingObject" />
<xs:enumeration value="loggingRecord" />
<xs:enumeration value="prescale" />
<xs:enumeration value="pulseRate" />
<xs:enumeration value="scale" />
<xs:enumeration value="scaleFactor" />
<xs:enumeration value="updateTime" />
<xs:enumeration value="valueBeforeChange" />
<xs:enumeration value="valueSet" />
<xs:enumeration value="valueChangeTime" />
<xs:enumeration value="alignIntervals" />
<xs:enumeration value="intervalOffset" />
<xs:enumeration value="lastRestartReason" />
<xs:enumeration value="loggingType" />
<xs:enumeration value="restartNotificationRecipients" />
<xs:enumeration value="timeOfDeviceRestart" />
<xs:enumeration value="timeSynchronizationInterval" />
<xs:enumeration value="trigger" />
<xs:enumeration value="utcTimeSynchronizationRecipients" />
<xs:enumeration value="nodeSubtype" />
<xs:enumeration value="nodeType" />
<xs:enumeration value="structuredObjectList" />
<xs:enumeration value="subordinateAnnotations" />
<xs:enumeration value="subordinateList" />
<xs:enumeration value="actualShedLevel" />
<xs:enumeration value="dutyWindow" />
<xs:enumeration value="expectedShedLevel" />
<xs:enumeration value="fullDutyBaseline" />
<xs:enumeration value="requestedShedLevel" />
<xs:enumeration value="shedDuration" />
<xs:enumeration value="shedLevelDescriptions" />
<xs:enumeration value="shedLevels" />
<xs:enumeration value="stateDescription" />
<xs:enumeration value="doorAlarmState" />
<xs:enumeration value="doorExtendedPulseTime" />

```



```

<xs:enumeration value="doorMembers" />
<xs:enumeration value="doorOpenTooLongTime" />
<xs:enumeration value="doorPulseTime" />
<xs:enumeration value="doorStatus" />
<xs:enumeration value="doorUnlockDelayTime" />
<xs:enumeration value="lockStatus" />
<xs:enumeration value="maskedAlarmValues" />
<xs:enumeration value="securedStatus" />
<xs:enumeration value="absenteeLimit" />
<xs:enumeration value="accessAlarmEvents" />
<xs:enumeration value="accessDoors" />
<xs:enumeration value="accessEvent" />
<xs:enumeration value="accessEventAuthenticationFactor" />
<xs:enumeration value="accessEventCredential" />
<xs:enumeration value="accessEventTime" />
<xs:enumeration value="accessTransactionEvents" />
<xs:enumeration value="accompaniment" />
<xs:enumeration value="accompanimentTime" />
<xs:enumeration value="activationTime" />
<xs:enumeration value="activeAuthenticationPolicy" />
<xs:enumeration value="assignedAccessRights" />
<xs:enumeration value="authenticationFactors" />
<xs:enumeration value="authenticationPolicyList" />
<xs:enumeration value="authenticationPolicyNames" />
<xs:enumeration value="authenticationStatus" />
<xs:enumeration value="authorizationMode" />
<xs:enumeration value="belongsTo" />
<xs:enumeration value="credentialDisable" />
<xs:enumeration value="credentialStatus" />
<xs:enumeration value="credentials" />
<xs:enumeration value="credentialsInZone" />
<xs:enumeration value="daysRemaining" />
<xs:enumeration value="entryPoints" />
<xs:enumeration value="exitPoints" />
<xs:enumeration value="expirationTime" />
<xs:enumeration value="extendedTimeEnable" />
<xs:enumeration value="failedAttemptEvents" />
<xs:enumeration value="failedAttempts" />
<xs:enumeration value="failedAttemptsTime" />
<xs:enumeration value="lastAccessEvent" />
<xs:enumeration value="lastAccessPoint" />
<xs:enumeration value="lastCredentialAdded" />
<xs:enumeration value="lastCredentialAddedTime" />
<xs:enumeration value="lastCredentialRemoved" />
<xs:enumeration value="lastCredentialRemovedTime" />
<xs:enumeration value="lastUseTime" />
<xs:enumeration value="lockout" />
<xs:enumeration value="lockoutRelinquishTime" />
<xs:enumeration value="maxFailedAttempts" />
<xs:enumeration value="members" />
<xs:enumeration value="musterPoint" />
<xs:enumeration value="negativeAccessRules" />
<xs:enumeration value="numberOfAuthenticationPolicies" />
<xs:enumeration value="occupancyCount" />
<xs:enumeration value="occupancyCountAdjust" />
<xs:enumeration value="occupancyCountEnable" />
<xs:enumeration value="occupancyLowerLimit" />
<xs:enumeration value="occupancyLowerLimitEnforced" />
<xs:enumeration value="occupancyState" />
<xs:enumeration value="occupancyUpperLimit" />
<xs:enumeration value="occupancyUpperLimitEnforced" />
<xs:enumeration value="passbackMode" />
<xs:enumeration value="passbackTimeout" />
<xs:enumeration value="positiveAccessRules" />
<xs:enumeration value="reasonForDisable" />
<xs:enumeration value="supportedFormats" />
<xs:enumeration value="supportedFormatClasses" />
<xs:enumeration value="threatAuthority" />
<xs:enumeration value="threatLevel" />
<xs:enumeration value="traceFlag" />
<xs:enumeration value="transactionNotificationClass" />

```

```

<xs:enumeration value="userExternalIdentifier" />
<xs:enumeration value="userInformationReference" />
<xs:enumeration value="userName" />
<xs:enumeration value="userType" />
<xs:enumeration value="usesRemaining" />
<xs:enumeration value="zoneFrom" />
<xs:enumeration value="zoneTo" />
<xs:enumeration value="accessEventTag" />
<xs:enumeration value="globalIdentifier" />
<xs:enumeration value="verificationTime" />
<xs:enumeration value="baseDeviceSecurityPolicy" />
<xs:enumeration value="distributionKeyRevision" />
<xs:enumeration value="doNotHide" />
<xs:enumeration value="keySets" />
<xs:enumeration value="lastKeyServer" />
<xs:enumeration value="networkAccessSecurityPolicies" />
<xs:enumeration value="packetReorderTime" />
<xs:enumeration value="securityPduTimeout" />
<xs:enumeration value="securityTimeWindow" />
<xs:enumeration value="supportedSecurityAlgorithms" />
<xs:enumeration value="updateKeySetTimeout" />
<xs:enumeration value="backupAndRestoreState" />
<xs:enumeration value="backupPreparationTime" />
<xs:enumeration value="restoreCompletionTime" />
<xs:enumeration value="restorePreparationTime" />
<xs:enumeration value="bitMask" />
<xs:enumeration value="bitText" />
<xs:enumeration value="isUtc" />
<xs:enumeration value="groupMembers" />
<xs:enumeration value="groupMemberNames" />
<xs:enumeration value="memberStatusFlags" />
<xs:enumeration value="requestedUpdateInterval" />
<xs:enumeration value="covuPeriod" />
<xs:enumeration value="covuRecipients" />
<xs:enumeration value="eventMessageTexts" />
<xs:enumeration value="eventMessageTextsConfig" />
<xs:enumeration value="eventDetectionEnable" />
<xs:enumeration value="eventAlgorithmInhibit" />
<xs:enumeration value="eventAlgorithmInhibitRef" />
<xs:enumeration value="timeDelayNormal" />
<xs:enumeration value="reliabilityEvaluationInhibit" />
<xs:enumeration value="faultParameters" />
<xs:enumeration value="faultType" />
<xs:enumeration value="localForwardingOnly" />
<xs:enumeration value="processIdentifierFilter" />
<xs:enumeration value="subscribedRecipients" />
<xs:enumeration value="portFilter" />
<xs:enumeration value="authorizationExemptions" />
<xs:enumeration value="allowGroupDelayInhibit" />
<xs:enumeration value="channelNumber" />
<xs:enumeration value="controlGroups" />
<xs:enumeration value="executionDelay" />
<xs:enumeration value="lastPriority" />
<xs:enumeration value="writeStatus" />
<xs:enumeration value="propertyList" />
<xs:enumeration value="serialNumber" />
<xs:enumeration value="blinkWarnEnable" />
<xs:enumeration value="defaultFadeTime" />
<xs:enumeration value="defaultRampRate" />
<xs:enumeration value="defaultStepIncrement" />
<xs:enumeration value="egressTime" />
<xs:enumeration value="inProgress" />
<xs:enumeration value="instantaneousPower" />
<xs:enumeration value="lightingCommand" />
<xs:enumeration value="lightingCommandDefaultPriority" />
<xs:enumeration value="maxActualValue" />
<xs:enumeration value="minActualValue" />
<xs:enumeration value="power" />
<xs:enumeration value="transition" />
<xs:enumeration value="egressActive" />
<xs:enumeration value="interfaceValue" />

```

```

<xs:enumeration value="faultHighLimit" />
<xs:enumeration value="faultLowLimit" />
<xs:enumeration value="lowDiffLimit" />
<xs:enumeration value="strikeCount" />
<xs:enumeration value="timeOfStrikeCountReset" />
<xs:enumeration value="defaultTimeout" />
<xs:enumeration value="initialTimeout" />
<xs:enumeration value="lastStateChange" />
<xs:enumeration value="stateChangeValues" />
<xs:enumeration value="timerRunning" />
<xs:enumeration value="timerState" />
<xs:enumeration value="apdulLength" />
<xs:enumeration value="ipAddress" />
<xs:enumeration value="ipDefaultGateway" />
<xs:enumeration value="ipDhcpEnable" />
<xs:enumeration value="ipDhcpLeaseTime" />
<xs:enumeration value="ipDhcpLeaseTimeRemaining" />
<xs:enumeration value="ipDhcpServer" />
<xs:enumeration value="ipDnsServer" />
<xs:enumeration value="bacnetIpGlobalAddress" />
<xs:enumeration value="bacnetIpMode" />
<xs:enumeration value="bacnetIpMulticastAddress" />
<xs:enumeration value="bacnetIpNatTraversal" />
<xs:enumeration value="ipSubnetMask" />
<xs:enumeration value="bacnetIpUdpPort" />
<xs:enumeration value="bbmdAcceptFdRegistrations" />
<xs:enumeration value="bbmdBroadcastDistributionTable" />
<xs:enumeration value="bbmdForeignDeviceTable" />
<xs:enumeration value="changesPending" />
<xs:enumeration value="command" />
<xs:enumeration value="fdBbmdAddress" />
<xs:enumeration value="fdSubscriptionLifetime" />
<xs:enumeration value="linkSpeed" />
<xs:enumeration value="linkSpeeds" />
<xs:enumeration value="linkSpeedAutonegotiate" />
<xs:enumeration value="macAddress" />
<xs:enumeration value="networkInterfaceName" />
<xs:enumeration value="networkNumber" />
<xs:enumeration value="networkNumberQuality" />
<xs:enumeration value="networkType" />
<xs:enumeration value="routingTable" />
<xs:enumeration value="virtualMacAddressTable" />
<xs:enumeration value="commandTimeArray" />
<xs:enumeration value="currentCommandPriority" />
<xs:enumeration value="lastCommandTime" />
<xs:enumeration value="valueSource" />
<xs:enumeration value="valueSourceArray" />
<xs:enumeration value="bacnetIpv6Mode" />
<xs:enumeration value="ipv6Address" />
<xs:enumeration value="ipv6PrefixLength" />
<xs:enumeration value="bacnetIpv6UdpPort" />
<xs:enumeration value="ipv6DefaultGateway" />
<xs:enumeration value="bacnetIpv6MulticastAddress" />
<xs:enumeration value="ipv6DnsServer" />
<xs:enumeration value="ipv6AutoAddressingEnable" />
<xs:enumeration value="ipv6DhcpLeaseTime" />
<xs:enumeration value="ipv6DhcpLeaseTimeRemaining" />
<xs:enumeration value="ipv6DhcpServer" />
<xs:enumeration value="ipv6ZoneIndex" />
<xs:enumeration value="assignedLandingCalls" />
<xs:enumeration value="carAssignedDirection" />
<xs:enumeration value="carDoorCommand" />
<xs:enumeration value="carDoorStatus" />
<xs:enumeration value="carDoorText" />
<xs:enumeration value="carDoorZone" />
<xs:enumeration value="carDriveStatus" />
<xs:enumeration value="carLoad" />
<xs:enumeration value="carLoadUnits" />
<xs:enumeration value="carMode" />
<xs:enumeration value="carMovingDirection" />
<xs:enumeration value="carPosition" />

```

```

<xs:enumeration value="elevatorGroup" />
<xs:enumeration value="energyMeter" />
<xs:enumeration value="energyMeterRef" />
<xs:enumeration value="escalatorMode" />
<xs:enumeration value="faultSignals" />
<xs:enumeration value="floorText" />
<xs:enumeration value="groupId" />
<xs:enumeration value="groupMode" />
<xs:enumeration value="higherDeck" />
<xs:enumeration value="installationId" />
<xs:enumeration value="landingCalls" />
<xs:enumeration value="landingCallControl" />
<xs:enumeration value="landingDoorStatus" />
<xs:enumeration value="lowerDeck" />
<xs:enumeration value="machineRoomId" />
<xs:enumeration value="makingCarCall" />
<xs:enumeration value="nextStoppingFloor" />
<xs:enumeration value="operationDirection" />
<xs:enumeration value="passengerAlarm" />
<xs:enumeration value="powerMode" />
<xs:enumeration value="registeredCarCall" />
<xs:enumeration value="activeCovMultipleSubscriptions" />
<xs:enumeration value="protocolLevel" />
<xs:enumeration value="referencePort" />
<xs:enumeration value="deployedProfileLocation" />
<xs:enumeration value="profileLocation" />
<xs:enumeration value="tags" />
<xs:enumeration value="subordinateNodeTypes" />
<xs:enumeration value="subordinateTags" />
<xs:enumeration value="subordinateRelationships" />
<xs:enumeration value="defaultSubordinateRelationship" />
<xs:enumeration value="represents" />
</xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<BACnetPropertyIdentifier>ackedTransitions</BACnetPropertyIdentifier>
```

BACnetPropertyReference

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetPropertyReference">
  <xs:sequence>
    <xs:element name="PropertyIdentifier" type="BACnetPropertyIdentifier" />
    <xs:element minOccurs="0" name="PropertyArrayIndex" type="Unsigned" />
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```

<BACnetPropertyReference>
  <PropertyIdentifier>ackedTransitions</PropertyIdentifier>
  <PropertyArrayIndex>1235</PropertyArrayIndex>
</BACnetPropertyReference>

```

Example 2:

```

<BACnetPropertyReference>
  <PropertyIdentifier>action</PropertyIdentifier>
</BACnetPropertyReference>

```

BACnetPropertyStates

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetPropertyStates">
  <xs:sequence>
    <xs:choice>
      <xs:element name="BooleanValue" type="Boolean" />
      <xs:element name="BinaryValue" type="BACnetBinaryPV" />
      <xs:element name="EventType" type="BACnetEventType" />
      <xs:element name="Polarity" type="BACnetPolarity" />
      <xs:element name="ProgramChange" type="BACnetProgramRequest" />
      <xs:element name="ProgramState" type="BACnetProgramState" />
      <xs:element name="ReasonForHalt" type="BACnetProgramError" />
      <xs:element name="Reliability" type="BACnetReliability" />
      <xs:element name="State" type="BACnetEventState" />
      <xs:element name="SystemStatus" type="BACnetDeviceStatus" />
      <xs:element name="Units" type="BACnetEngineeringUnits" />
      <xs:element name="UnsignedValue" type="Unsigned" />
      <xs:element name="LifeSafetyMode" type="BACnetLifeSafetyMode" />
      <xs:element name="LifeSafetyState" type="BACnetLifeSafetyState" />
      <xs:element name="RestartReason" type="BACnetRestartReason" />
      <xs:element name="DoorAlarmState" type="BACnetDoorAlarmState" />
      <xs:element name="Action" type="BACnetAction" />
      <xs:element name="DoorSecuredStatus" type="BACnetDoorSecuredStatus" />
      <xs:element name="DoorStatus" type="BACnetDoorStatus" />
      <xs:element name="DoorValue" type="BACnetDoorValue" />
      <xs:element name="FileAccessMethod" type="BACnetFileAccessMethod" />
      <xs:element name="LockStatus" type="BACnetLockStatus" />
      <xs:element name="LifeSafetyOperation" type="BACnetLifeSafetyOperation" />
      <xs:element name="Maintenance" type="BACnetMaintenance" />
      <xs:element name="NodeType" type="BACnetNodeType" />
      <xs:element name="NotifyType" type="BACnetNotifyType" />
      <xs:element name="SecurityLevel" type="BACnetSecurityLevel" />
      <xs:element name="ShedState" type="BACnetShedState" />
      <xs:element name="SilencedState" type="BACnetSilencedState" />
      <xs:element name="AccessEvent" type="BACnetAccessEvent" />
      <xs:element name="ZoneOccupancyState" type="BACnetAccessZoneOccupancyState" />
      <xs:element name="AccessCredentialDisableReason" type="BACnetAccessCredentialDisableReason" />
      <xs:element name="AccessCredentialDisable" type="BACnetAccessCredentialDisable" />
      <xs:element name="AuthenticationStatus" type="BACnetAuthenticationStatus" />
      <xs:element name="BackupState" type="BACnetBackupState" />
      <xs:element name="WriteStatus" type="BACnetWriteStatus" />
      <xs:element name="LightingInProgress" type="BACnetLightingInProgress" />
      <xs:element name="LightingOperation" type="BACnetLightingOperation" />
      <xs:element name="LightingTransition" type="BACnetLightingTransition" />
      <xs:element name="IntegerValue" type="INTEGER" />
      <xs:element name="BinaryLightingValue" type="BACnetBinaryLightingPV" />
      <xs:element name="TimerState" type="BACnetTimerState" />
      <xs:element name="TimerTransition" type="BACnetTimerTransition" />
      <xs:element name="BacnetIpMode" type="BACnetIPMode" />
      <xs:element name="NetworkPortCommand" type="BACnetNetworkPortCommand" />
      <xs:element name="NetworkType" type="BACnetNetworkType" />
      <xs:element name="NetworkNumberQuality" type="BACnetNetworkNumberQuality" />
      <xs:element name="EscalatorOperationDirection" type="BACnetEscalatorOperationDirection" />
      <xs:element name="EscalatorFault" type="BACnetEscalatorFault" />
      <xs:element name="EscalatorMode" type="BACnetEscalatorMode" />
      <xs:element name="LiftCarDirection" type="BACnetLiftCarDirection" />
      <xs:element name="LiftCarDoorCommand" type="BACnetLiftCarDoorCommand" />
      <xs:element name="LiftCarDriveStatus" type="BACnetLiftCarDriveStatus" />
      <xs:element name="LiftCarMode" type="BACnetLiftCarMode" />
      <xs:element name="LiftGroupMode" type="BACnetLiftGroupMode" />
      <xs:element name="LiftFault" type="BACnetLiftFault" />
      <xs:element name="ProtocolLevel" type="BACnetProtocolLevel" />
      <xs:element name="ExtendedValue" type="Unsigned32" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```
<BACnetPropertyStates>
  <BooleanValue>true</BooleanValue>
</BACnetPropertyStates>
```

Example 2:

```
<BACnetPropertyStates>
  <BinaryValue>inactive</BinaryValue>
</BACnetPropertyStates>
```

Example 3:

```
<BACnetPropertyStates>
  <EventType>changeOfState</EventType>
</BACnetPropertyStates>
```

Example 4:

```
<BACnetPropertyStates>
  <Polarity>normal</Polarity>
</BACnetPropertyStates>
```

Example 5:

```
<BACnetPropertyStates>
  <ProgramChange>halt</ProgramChange>
</BACnetPropertyStates>
```

Example 6:

```
<BACnetPropertyStates>
  <ProgramState>halted</ProgramState>
</BACnetPropertyStates>
```

Example 7:

```
<BACnetPropertyStates>
  <ReasonForHalt>normal</ReasonForHalt>
</BACnetPropertyStates>
```

Example 8:

```
<BACnetPropertyStates>
  <Reliability>noOutput</Reliability>
</BACnetPropertyStates>
```

Example 9:

```
<BACnetPropertyStates>
  <State>fault</State>
</BACnetPropertyStates>
```

Example 10:

```
<BACnetPropertyStates>
  <SystemStatus>downloadRequired</SystemStatus>
</BACnetPropertyStates>
```

Example 11:

```
<BACnetPropertyStates>
  <Units>kilovoltAmperes</Units>
</BACnetPropertyStates>
```

Example 12:

```
<BACnetPropertyStates>
  <UnsignedValue>1244</UnsignedValue>
</BACnetPropertyStates>
```

Example 13:

```
<BACnetPropertyStates>
  <LifeSafetyMode>enabled</LifeSafetyMode>
</BACnetPropertyStates>
```

Example 14:

```
<BACnetPropertyStates>
  <LifeSafetyState>testFaultAlarm</LifeSafetyState>
</BACnetPropertyStates>
```

Example 15:

```
<BACnetPropertyStates>
  <RestartReason>detectedPoweredOff</RestartReason>
</BACnetPropertyStates>
```

Example 16:

```
<BACnetPropertyStates>
  <DoorAlarmState>doorFault</DoorAlarmState>
</BACnetPropertyStates>
```

Example 17:

```
<BACnetPropertyStates>
  <Action>reverse</Action>
</BACnetPropertyStates>
```

Example 18:

```
<BACnetPropertyStates>
  <DoorSecuredStatus>unsecured</DoorSecuredStatus>
</BACnetPropertyStates>
```

Example 19:

```
<BACnetPropertyStates>
  <DoorStatus>opening</DoorStatus>
</BACnetPropertyStates>
```

Example 20:

```
<BACnetPropertyStates>
  <DoorValue>pulseUnlock</DoorValue>
</BACnetPropertyStates>
```

Example 21:

```
<BACnetPropertyStates>
  <FileAccessMethod>streamAccess</FileAccessMethod>
</BACnetPropertyStates>
```

Example 22:

```
<BACnetPropertyStates>
  <LockStatus>locked</LockStatus>
</BACnetPropertyStates>
```

Example 23:

```
<BACnetPropertyStates>
  <LifeSafetyOperation>silence</LifeSafetyOperation>
</BACnetPropertyStates>
```

Example 24:

```
<BACnetPropertyStates>
  <Maintenance>needServiceOperational</Maintenance>
</BACnetPropertyStates>
```

Example 25:

```
<BACnetPropertyStates>
  <NodeType>system</NodeType>
</BACnetPropertyStates>
```

Example 26:

```
<BACnetPropertyStates>
  <NotifyType>alarm</NotifyType>
</BACnetPropertyStates>
```

Example 27:


```
<BACnetPropertyStates>
  <SecurityLevel>plain</SecurityLevel>
</BACnetPropertyStates>
```

Example 28:

```
<BACnetPropertyStates>
  <ShedState>shedCompliant</ShedState>
</BACnetPropertyStates>
```

Example 29:

```
<BACnetPropertyStates>
  <SilencedState>allSilenced</SilencedState>
</BACnetPropertyStates>
```

Example 30:

```
<BACnetPropertyStates>
  <AccessEvent>deniedUnexpectedLocationUsage</AccessEvent>
</BACnetPropertyStates>
```

Example 31:

```
<BACnetPropertyStates>
  <ZoneOccupancyState>belowLowerLimit</ZoneOccupancyState>
</BACnetPropertyStates>
```

Example 32:

```
<BACnetPropertyStates>
  <AccessCredentialDisableReason>disabled</AccessCredentialDisableReason>
</BACnetPropertyStates>
```

Example 33:

```
<BACnetPropertyStates>
  <AccessCredentialDisable>disableLockout</AccessCredentialDisable>
</BACnetPropertyStates>
```

Example 34:

```
<BACnetPropertyStates>
  <AuthenticationStatus>waitingForAccompaniment</AuthenticationStatus>
</BACnetPropertyStates>
```

Example 35:

```
<BACnetPropertyStates>
  <BackupState>backupFailure</BackupState>
</BACnetPropertyStates>
```

Example 36:

```
<BACnetPropertyStates>
  <WriteStatus>successful</WriteStatus>
</BACnetPropertyStates>
```

Example 37:

```
<BACnetPropertyStates>
  <LightingInProgress>idle</LightingInProgress>
</BACnetPropertyStates>
```

Example 38:

```
<BACnetPropertyStates>
  <LightingOperation>stepUp</LightingOperation>
</BACnetPropertyStates>
```

Example 39:

```
<BACnetPropertyStates>
  <LightingTransition>fade</LightingTransition>
</BACnetPropertyStates>
```

Example 40:

```
<BACnetPropertyStates>
  <IntegerValue>-762</IntegerValue>
</BACnetPropertyStates>
```

Example 41:

```
<BACnetPropertyStates>
  <BinaryLightingValue>warnOff</BinaryLightingValue>
</BACnetPropertyStates>
```

Example 42:

```
<BACnetPropertyStates>
  <TimerState>running</TimerState>
</BACnetPropertyStates>
```

Example 43:

```
<BACnetPropertyStates>
  <TimerTransition>idleToRunning</TimerTransition>
</BACnetPropertyStates>
```

Example 44:

```
<BACnetPropertyStates>
  <BacnetIpMode>normal</BacnetIpMode>
</BACnetPropertyStates>
```

Example 45:

```
<BACnetPropertyStates>
  <NetworkPortCommand>restartSlaveDiscovery</NetworkPortCommand>
</BACnetPropertyStates>
```

Example 46:

```
<BACnetPropertyStates>
  <NetworkType>lontalk</NetworkType>
</BACnetPropertyStates>
```

Example 47:

```
<BACnetPropertyStates>
  <NetworkNumberQuality>learned</NetworkNumberQuality>
</BACnetPropertyStates>
```

Example 48:

```
<BACnetPropertyStates>
  <EscalatorOperationDirection>downRatedSpeed</EscalatorOperationDirection>
</BACnetPropertyStates>
```

Example 49:

```
<BACnetPropertyStates>
  <EscalatorFault>mechanicalComponentFault</EscalatorFault>
</BACnetPropertyStates>
```

Example 50:

```
<BACnetPropertyStates>
  <EscalatorMode>unknown</EscalatorMode>
</BACnetPropertyStates>
```

Example 51:

```
<BACnetPropertyStates>
  <LiftCarDirection>none</LiftCarDirection>
</BACnetPropertyStates>
```

Example 52:

```
<BACnetPropertyStates>
  <LiftCarDoorCommand>close</LiftCarDoorCommand>
</BACnetPropertyStates>
```

Example 53:

```
<BACnetPropertyStates>
  <LiftCarDriveStatus>stationary</LiftCarDriveStatus>
</BACnetPropertyStates>
```

Example 54:

```
<BACnetPropertyStates>
  <LiftCarMode>earthquakeOperation</LiftCarMode>
</BACnetPropertyStates>
```

Example 55:

```
<BACnetPropertyStates>
  <LiftGroupMode>fourWay</LiftGroupMode>
</BACnetPropertyStates>
```

Example 56:

```
<BACnetPropertyStates>
  <LiftFault>liftShaftDeviceFault</LiftFault>
</BACnetPropertyStates>
```

Example 57:

```
<BACnetPropertyStates>
  <ProtocolLevel>nonBacnetApplication</ProtocolLevel>
</BACnetPropertyStates>
```

Example 58:

```
<BACnetPropertyStates>
  <ExtendedValue>2401</ExtendedValue>
</BACnetPropertyStates>
```

BACnetPropertyValue

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetPropertyValue">
  <xs:sequence>
    <xs:element name="PropertyIdentifier" type="BACnetPropertyIdentifier" />
    <xs:element minOccurs="0" name="PropertyArrayIndex" type="Unsigned" />
    <xs:element name="PropertyValue" type="Any" />
    <xs:element minOccurs="0" name="Priority" type="Unsigned" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetPropertyValue>
  <PropertyIdentifier>ackedTransitions</PropertyIdentifier>
  <PropertyArrayIndex>1235</PropertyArrayIndex>
  <PropertyValue>
    <Boolean>true</Boolean>
  </PropertyValue>
  <Priority>1236</Priority>
</BACnetPropertyValue>
```

Example 2:

```
<BACnetPropertyValue>
  <PropertyIdentifier>actionText</PropertyIdentifier>
  <PropertyValue>
    <Boolean>true</Boolean>
  </PropertyValue>
</BACnetPropertyValue>
```

BACnetProtocolLevel

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetProtocolLevel">
  <xs:restriction base="xs:string">
    <xs:enumeration value="physical" />
    <xs:enumeration value="protocol" />
    <xs:enumeration value="bacnetApplication" />
    <xs:enumeration value="nonBacnetApplication" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetProtocolLevel>physical</BACnetProtocolLevel>
```

BACnetRecipient

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetRecipient">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Device" type="BACnetObjectIdentifier" />
      <xs:element name="Address" type="BACnetAddress" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetRecipient>
  <Device>device-501</Device>
</BACnetRecipient>
```

Example 2:

```
<BACnetRecipient>
  <Address>
    <NetworkNumber>7635</NetworkNumber>
    <MacAddress>FF</MacAddress>
  </Address>
</BACnetRecipient>
```

BACnetRecipientProcess

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetRecipientProcess">
  <xs:sequence>
    <xs:element name="Recipient" type="BACnetRecipient" />
    <xs:element name="ProcessIdentifier" type="Unsigned32" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetRecipientProcess>
  <Recipient>
    <Device>device-501</Device>
  </Recipient>
  <ProcessIdentifier>2346</ProcessIdentifier>
</BACnetRecipientProcess>
```

BACnetRelationship

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetRelationship">
  <xs:restriction base="xs:string">
    <xs:enumeration value="unknown" />
    <xs:enumeration value="default" />
    <xs:enumeration value="contains" />
    <xs:enumeration value="containedBy" />
    <xs:enumeration value="uses" />
    <xs:enumeration value="usedBy" />
    <xs:enumeration value="commands" />
    <xs:enumeration value="commandedBy" />
    <xs:enumeration value="adjusts" />
    <xs:enumeration value="adjustedBy" />
    <xs:enumeration value="ingress" />
    <xs:enumeration value="egress" />
    <xs:enumeration value="suppliesAir" />
    <xs:enumeration value="receivesAir" />
    <xs:enumeration value="suppliesHotAir" />
    <xs:enumeration value="receivesHotAir" />
    <xs:enumeration value="suppliesCoolAir" />
    <xs:enumeration value="receivesCoolAir" />
    <xs:enumeration value="suppliesPower" />
    <xs:enumeration value="receivesPower" />
    <xs:enumeration value="suppliesGas" />
    <xs:enumeration value="receivesGas" />
    <xs:enumeration value="suppliesWater" />
    <xs:enumeration value="receivesWater" />
    <xs:enumeration value="suppliesHotWater" />
    <xs:enumeration value="receivesHotWater" />
    <xs:enumeration value="suppliesCoolWater" />
    <xs:enumeration value="receivesCoolWater" />
    <xs:enumeration value="suppliesSteam" />
    <xs:enumeration value="receivesSteam" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetRelationship>unknown</BACnetRelationship>
```

BACnetReliability

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:simpleType name="BACnetReliability">
  <xs:restriction base="xs:string">
    <xs:enumeration value="noFaultDetected" />
    <xs:enumeration value="noSensor" />
    <xs:enumeration value="overRange" />
    <xs:enumeration value="underRange" />
    <xs:enumeration value="openLoop" />
    <xs:enumeration value="shortedLoop" />
    <xs:enumeration value="noOutput" />
    <xs:enumeration value="unreliableOther" />
    <xs:enumeration value="processError" />
    <xs:enumeration value="multiStateFault" />
    <xs:enumeration value="configurationError" />
    <xs:enumeration value="communicationFailure" />
    <xs:enumeration value="memberFault" />
    <xs:enumeration value="monitoredObjectFault" />
    <xs:enumeration value="tripped" />
    <xs:enumeration value="lampFailure" />
    <xs:enumeration value="activationFailure" />
    <xs:enumeration value="renewDhcpFailure" />
    <xs:enumeration value="renewFdRegistrationFailure" />
    <xs:enumeration value="restartAutoNegotiationFailure" />
    <xs:enumeration value="restartFailure" />
    <xs:enumeration value="proprietaryCommandFailure" />
    <xs:enumeration value="faultsListed" />
    <xs:enumeration value="referencedObjectFault" />
  </xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<BACnetReliability>noFaultDetected</BACnetReliability>
```

BACnetRestartReason

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:simpleType name="BACnetRestartReason">
  <xs:restriction base="xs:string">
    <xs:enumeration value="unknown" />
    <xs:enumeration value="coldstart" />
    <xs:enumeration value="warmstart" />
    <xs:enumeration value="detectedPowerLost" />
    <xs:enumeration value="detectedPoweredOff" />
    <xs:enumeration value="hardwareWatchdog" />
    <xs:enumeration value="softwareWatchdog" />
    <xs:enumeration value="suspended" />
    <xs:enumeration value="activateChanges" />
  </xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<BACnetRestartReason>unknown</BACnetRestartReason>
```

BACnetRouterEntry

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetRouterEntry">
  <xs:sequence>
    <xs:element name="NetworkNumber" type="Unsigned16" />
    <xs:element name="MacAddress" type="OctetString" />
    <xs:element name="Status" type="BACnetRouterEntry_Status" />
    <xs:element minOccurs="0" name="PerformanceIndex" type="Unsigned8" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetRouterEntry>
  <NetworkNumber>7634</NetworkNumber>
  <MacAddress>00010203040506070809</MacAddress>
  <Status>busy</Status>
  <PerformanceIndex>35</PerformanceIndex>
</BACnetRouterEntry>
```

Example 2:

```
<BACnetRouterEntry>
  <NetworkNumber>7637</NetworkNumber>
  <MacAddress>FF</MacAddress>
  <Status>busy</Status>
</BACnetRouterEntry>
```

BACnetRouterEntry_Status

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetRouterEntry_Status">
  <xs:restriction base="xs:string">
    <xs:enumeration value="available" />
    <xs:enumeration value="busy" />
    <xs:enumeration value="disconnected" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetRouterEntry_Status>available</BACnetRouterEntry_Status>
```

BACnetSecurityKeySet

This XML representation of complex data is used by the following objects/properties:

- **networkSecurity.keySets**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetSecurityKeySet">
    <xs:sequence>
      <xs:element name="KeyRevision" type="Unsigned8" />
      <xs:element name="ActivationTime" type="BACnetDateTime" />
      <xs:element name="ExpirationTime" type="BACnetDateTime" />
      <xs:element name="KeyIds" type="SequenceOfBACnetKeyIdentifier" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetSecurityKeySet" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetSecurityKeySet>
    <KeyRevision>33</KeyRevision>
    <ActivationTime>
      <Date>2018-12-20</Date>
      <Time>22:39:15.980</Time>
    </ActivationTime>
    <ExpirationTime>
      <Date>2019-12-20</Date>
      <Time>22:39:15.980</Time>
    </ExpirationTime>
    <KeyIds />
  </BACnetSecurityKeySet>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <BACnetSecurityKeySet>
    <KeyRevision>36</KeyRevision>
    <ActivationTime>
      <Date>2021-12-20</Date>
      <Time>22:39:15.980</Time>
    </ActivationTime>
    <ExpirationTime>
      <Date>2022-12-20</Date>
      <Time>22:39:15.980</Time>
    </ExpirationTime>
    <KeyIds>
      <BACnetKeyIdentifier>
        <Algorithm>39</Algorithm>
        <KeyId>40</KeyId>
      </BACnetKeyIdentifier>
      <BACnetKeyIdentifier>
        <Algorithm>41</Algorithm>
        <KeyId>42</KeyId>
      </BACnetKeyIdentifier>
      <BACnetKeyIdentifier>
        <Algorithm>43</Algorithm>
        <KeyId>44</KeyId>
      </BACnetKeyIdentifier>
    </KeyIds>
  </BACnetSecurityKeySet>
</Value>
```

BACnetSecurityLevel

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetSecurityLevel">
  <xs:restriction base="xs:string">
    <xs:enumeration value="incapable" />
    <xs:enumeration value="plain" />
    <xs:enumeration value="signed" />
    <xs:enumeration value="encrypted" />
    <xs:enumeration value="signedEndToEnd" />
    <xs:enumeration value="encryptedEndToEnd" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetSecurityLevel>incapable</BACnetSecurityLevel>
```

BACnetSecurityPolicy

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetSecurityPolicy">
  <xs:restriction base="xs:string">
    <xs:enumeration value="plainNonTrusted" />
    <xs:enumeration value="plainTrusted" />
    <xs:enumeration value="signedTrusted" />
    <xs:enumeration value="encryptedTrusted" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetSecurityPolicy>plainNonTrusted</BACnetSecurityPolicy>
```

BACnetShedState

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetShedState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="shedInactive" />
    <xs:enumeration value="shedRequestPending" />
    <xs:enumeration value="shedCompliant" />
    <xs:enumeration value="shedNonCompliant" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetShedState>shedInactive</BACnetShedState>
```

BACnetSilencedState

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetSilencedState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="unsilenced" />
    <xs:enumeration value="audibleSilenced" />
    <xs:enumeration value="visibleSilenced" />
    <xs:enumeration value="allSilenced" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetSilencedState>unsilenced</BACnetSilencedState>
```

BACnetSpecialEvent

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetSpecialEvent">
  <xs:sequence>
    <xs:element name="Period" type="BACnetSpecialEvent_Period" />
    <xs:element name="ListOfTimeValues" type="SequenceOfBACnetTimeValue" />
    <xs:element name="EventPriority" type="Unsigned" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetSpecialEvent>
  <Period>
    <CalendarEntry>
      <Date>2017-12-20</Date>
    </CalendarEntry>
  </Period>
  <ListOfTimeValues />
  <EventPriority>1235</EventPriority>
</BACnetSpecialEvent>
```

Example 2:

```

<BACnetSpecialEvent>
  <Period>
    <CalendarEntry>
      <Date>2019-12-20</Date>
    </CalendarEntry>
  </Period>
  <ListOfTimeValues>
    <BACnetTimeValue>
      <Time>22:39:15.980</Time>
      <Value>
        <Boolean>true</Boolean>
      </Value>
    </BACnetTimeValue>
    <BACnetTimeValue>
      <Time>22:39:15.980</Time>
      <Value>
        <Boolean>true</Boolean>
      </Value>
    </BACnetTimeValue>
    <BACnetTimeValue>
      <Time>22:39:15.980</Time>
      <Value>
        <Boolean>true</Boolean>
      </Value>
    </BACnetTimeValue>
  </ListOfTimeValues>
  <EventPriority>1240</EventPriority>
</BACnetSpecialEvent>

```

BACnetSpecialEvent_Period

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="BACnetSpecialEvent_Period">
  <xs:sequence>
    <xs:choice>
      <xs:element name="CalendarEntry" type="BACnetCalendarEntry" />
      <xs:element name="CalendarReference" type="BACnetObjectIdentifier" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```

<BACnetSpecialEvent_Period>
  <CalendarEntry>
    <Date>2017-12-20</Date>
  </CalendarEntry>
</BACnetSpecialEvent_Period>

```

Example 2:

```

<BACnetSpecialEvent_Period>
  <CalendarReference>device-502</CalendarReference>
</BACnetSpecialEvent_Period>

```

BACnetStatusFlags

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetStatusFlags">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="inAlarm" type="Boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="fault" type="Boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="overridden" type="Boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="outOfService" type="Boolean" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetStatusFlags>
  <inAlarm>true</inAlarm>
  <overridden>true</overridden>
</BACnetStatusFlags>
```

BACnetTimeStamp

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.commandTimeArray**
- **accessDoor.eventTimeStamps**
- **accessDoor.lastCommandTime**
- **accessPoint.accessEventTime**
- **accessPoint.eventTimeStamps**
- **accessZone.eventTimeStamps**
- **accumulator.eventTimeStamps**
- **alertEnrollment.eventTimeStamps**
- **analogInput.eventTimeStamps**
- **analogOutput.commandTimeArray**
- **analogOutput.eventTimeStamps**
- **analogOutput.lastCommandTime**
- **analogValue.commandTimeArray**
- **analogValue.eventTimeStamps**
- **analogValue.lastCommandTime**
- **binaryInput.eventTimeStamps**
- **binaryLightingOutput.commandTimeArray**
- **binaryLightingOutput.eventTimeStamps**
- **binaryLightingOutput.lastCommandTime**
- **binaryOutput.commandTimeArray**
- **binaryOutput.eventTimeStamps**
- **binaryOutput.lastCommandTime**

- `binaryValue.commandTimeArray`
- `binaryValue.eventTimeStamps`
- `binaryValue.lastCommandTime`
- `bitstringValue.commandTimeArray`
- `bitstringValue.eventTimeStamps`
- `bitstringValue.lastCommandTime`
- `channel.eventTimeStamps`
- `characterstringValue.commandTimeArray`
- `characterstringValue.eventTimeStamps`
- `characterstringValue.lastCommandTime`
- `command.eventTimeStamps`
- `credentialDataInput.eventTimeStamps`
- `credentialDataInput.updateTime`
- `dateValue.commandTimeArray`
- `dateValue.eventTimeStamps`
- `dateValue.lastCommandTime`
- `datepatternValue.commandTimeArray`
- `datepatternValue.eventTimeStamps`
- `datepatternValue.lastCommandTime`
- `datetimeValue.commandTimeArray`
- `datetimeValue.eventTimeStamps`
- `datetimeValue.lastCommandTime`
- `datetimepatternValue.commandTimeArray`
- `datetimepatternValue.eventTimeStamps`
- `datetimepatternValue.lastCommandTime`
- `device.eventTimeStamps`
- `device.lastRestoreTime`
- `device.timeOfDeviceRestart`
- `escalator.eventTimeStamps`
- `eventEnrollment.eventTimeStamps`
- `eventLog.eventTimeStamps`
- `globalGroup.eventTimeStamps`

- `integerValue.commandTimeArray`
- `integerValue.eventTimeStamps`
- `integerValue.lastCommandTime`
- `largeAnalogValue.commandTimeArray`
- `largeAnalogValue.eventTimeStamps`
- `largeAnalogValue.lastCommandTime`
- `lifeSafetyPoint.eventTimeStamps`
- `lifeSafetyZone.eventTimeStamps`
- `lift.eventTimeStamps`
- `lightingOutput.commandTimeArray`
- `lightingOutput.lastCommandTime`
- `loadControl.eventTimeStamps`
- `loop.eventTimeStamps`
- `multiStateInput.eventTimeStamps`
- `multiStateOutput.commandTimeArray`
- `multiStateOutput.eventTimeStamps`
- `multiStateOutput.lastCommandTime`
- `multiStateValue.commandTimeArray`
- `multiStateValue.eventTimeStamps`
- `multiStateValue.lastCommandTime`
- `networkPort.eventTimeStamps`
- `notificationClass.eventTimeStamps`
- `octetstringValue.commandTimeArray`
- `octetstringValue.lastCommandTime`
- `positiveIntegerValue.commandTimeArray`
- `positiveIntegerValue.eventTimeStamps`
- `positiveIntegerValue.lastCommandTime`
- `program.eventTimeStamps`
- `pulseConverter.eventTimeStamps`
- `schedule.eventTimeStamps`
- `timeValue.commandTimeArray`
- `timeValue.eventTimeStamps`

- **timeValue.lastCommandTime**
- **timepatternValue.commandTimeArray**
- **timepatternValue.eventTimeStamps**
- **timepatternValue.lastCommandTime**
- **timer.eventTimeStamps**
- **trendLog.eventTimeStamps**
- **trendLogMultiple.eventTimeStamps**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="BACnetTimeStamp">
    <xs:sequence>
      <xs:choice>
        <xs:element name="Time" type="Time" />
        <xs:element name="SequenceNumber" type="Unsigned" />
        <xs:element name="Datetime" type="BACnetDateTime" />
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="BACnetTimeStamp" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetTimeStamp>
    <Time>22:39:15.980</Time>
  </BACnetTimeStamp>
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetTimeStamp>
    <SequenceNumber>1235</SequenceNumber>
  </BACnetTimeStamp>
</Value>
```

Example 3:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BACnetTimeStamp>
    <Datetime>
      <Date>2019-12-20</Date>
      <Time>22:39:15.980</Time>
    </Datetime>
  </BACnetTimeStamp>
</Value>
```

BACnetTimeValue

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetTimeValue">
  <xs:sequence>
    <xs:element name="Time" type="Time" />
    <xs:element name="Value" type="Any" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetTimeValue>
  <Time>22:39:15.980</Time>
  <Value>
    <Boolean>true</Boolean>
  </Value>
</BACnetTimeValue>
```

BACnetTimerState

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetTimerState">
  <xs:restriction base="xs:string">
    <xs:enumeration value="idle" />
    <xs:enumeration value="running" />
    <xs:enumeration value="expired" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetTimerState>idle</BACnetTimerState>
```

BACnetTimerTransition

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetTimerTransition">
  <xs:restriction base="xs:string">
    <xs:enumeration value="none" />
    <xs:enumeration value="idleToRunning" />
    <xs:enumeration value="runningToIdle" />
    <xs:enumeration value="runningToRunning" />
    <xs:enumeration value="runningToExpired" />
    <xs:enumeration value="forcedToExpired" />
    <xs:enumeration value="expiredToIdle" />
    <xs:enumeration value="expiredToRunning" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetTimerTransition>none</BACnetTimerTransition>
```

BACnetVMACEntry

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema


```
<xs:complexType name="BACnetVMACEntry">
  <xs:sequence>
    <xs:element name="VirtualMacAddress" type="OctetString" />
    <xs:element name="NativeMacAddress" type="OctetString" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetVMACEntry>
  <VirtualMacAddress>00010203040506070809</VirtualMacAddress>
  <NativeMacAddress>00010203040506070809</NativeMacAddress>
</BACnetVMACEntry>
```

BACnetVTClass

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetVTClass">
  <xs:restriction base="xs:string">
    <xs:enumeration value="defaultTerminal" />
    <xs:enumeration value="ansiX364" />
    <xs:enumeration value="decVt52" />
    <xs:enumeration value="decVt100" />
    <xs:enumeration value="decVt220" />
    <xs:enumeration value="hp70094" />
    <xs:enumeration value="ibm3130" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetVTClass>defaultTerminal</BACnetVTClass>
```

BACnetVTSession

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="BACnetVTSession">
  <xs:sequence>
    <xs:element name="LocalVtSessionId" type="Unsigned8" />
    <xs:element name="RemoteVtSessionId" type="Unsigned8" />
    <xs:element name="RemoteVtAddress" type="BACnetAddress" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<BACnetVTSession>
  <LocalVtSessionId>33</LocalVtSessionId>
  <RemoteVtSessionId>34</RemoteVtSessionId>
  <RemoteVtAddress>
    <NetworkNumber>7636</NetworkNumber>
    <MacAddress>00010203040506070809</MacAddress>
  </RemoteVtAddress>
</BACnetVTSession>
```

BACnetWriteStatus

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="BACnetWriteStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="idle" />
    <xs:enumeration value="inProgress" />
    <xs:enumeration value="successful" />
    <xs:enumeration value="failed" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<BACnetWriteStatus>idle</BACnetWriteStatus>
```

BitString

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.priorityArray**
- **analogOutput.priorityArray**
- **analogValue.priorityArray**
- **binaryLightingOutput.priorityArray**
- **binaryOutput.priorityArray**
- **binaryValue.priorityArray**
- **bitstringValue.priorityArray**
- **channel.presentValue**
- **characterstringValue.priorityArray**
- **dateValue.priorityArray**
- **datepatternValue.priorityArray**
- **datetimeValue.priorityArray**
- **datetimepatternValue.priorityArray**
- **integerValue.priorityArray**
- **largeAnalogValue.priorityArray**
- **lightingOutput.priorityArray**
- **multiStateOutput.priorityArray**
- **multiStateValue.priorityArray**
- **octetstringValue.priorityArray**
- **positiveIntegerValue.priorityArray**
- **timeValue.priorityArray**
- **timepatternValue.priorityArray**

- **timer.stateChangeValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="BitString">
    <xs:restriction base="xs:string" />
  </xs:simpleType>
  <xs:element name="Value" type="BitString" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <BitString>101</BitString>
</Value>
```

Boolean

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.priorityArray**
- **analogOutput.priorityArray**
- **analogValue.priorityArray**
- **binaryLightingOutput.priorityArray**
- **binaryOutput.priorityArray**
- **binaryValue.priorityArray**
- **bitstringValue.priorityArray**
- **channel.presentValue**
- **characterstringValue.priorityArray**
- **dateValue.priorityArray**
- **datepatternValue.priorityArray**
- **datetimeValue.priorityArray**
- **datetimepatternValue.priorityArray**
- **integerValue.priorityArray**
- **largeAnalogValue.priorityArray**
- **lightingOutput.priorityArray**
- **multiStateOutput.priorityArray**
- **multiStateValue.priorityArray**
- **octetstringValue.priorityArray**

- **positiveIntegerValue.priorityArray**
- **timeValue.priorityArray**
- **timepatternValue.priorityArray**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="Boolean">
    <xs:restriction base="xs:boolean" />
  </xs:simpleType>
  <xs:element name="Value" type="Boolean" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <Boolean>true</Boolean>
</Value>
```

CharacterString

This XML representation of complex data is used by the following objects/properties:

- **networkPort.bacnetIpGlobalAddress**
- **networkPort.fdBbmdAddress**
- **timer.stateChangeValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="CharacterString">
    <xs:restriction base="xs:string" />
  </xs:simpleType>
  <xs:element name="Value" type="CharacterString" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <CharacterString>Abcde</CharacterString>
</Value>
```

ConfirmedEventNotificationRequest

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="ConfirmedEventNotificationRequest">
  <xs:sequence>
    <xs:element name="ProcessIdentifier" type="Unsigned32" />
    <xs:element name="InitiatingDeviceIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="EventObjectIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="Timestamp" type="BACnetTimeStamp" />
    <xs:element name="NotificationClass" type="Unsigned" />
    <xs:element name="Priority" type="Unsigned8" />
    <xs:element name="EventType" type="BACnetEventType" />
    <xs:element minOccurs="0" name="MessageText" type="CharacterString" />
    <xs:element name="NotifyType" type="BACnetNotifyType" />
    <xs:element minOccurs="0" name="AckRequired" type="Boolean" />
    <xs:element minOccurs="0" name="FromState" type="BACnetEventState" />
    <xs:element name="ToState" type="BACnetEventState" />
    <xs:element minOccurs="0" name="EventValues" type="BACnetNotificationParameters" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<ConfirmedEventNotificationRequest>
  <ProcessIdentifier>2345</ProcessIdentifier>
  <InitiatingDeviceIdentifier>device-502</InitiatingDeviceIdentifier>
  <EventObjectIdentifier>device-503</EventObjectIdentifier>
  <Timestamp>
    <Time>22:39:15.980</Time>
  </Timestamp>
  <NotificationClass>1238</NotificationClass>
  <Priority>38</Priority>
  <EventType>changeOfLifeSafety</EventType>
  <MessageText>Abcde</MessageText>
  <NotifyType>event</NotifyType>
  <AckRequired>true</AckRequired>
  <FromState>offnormal</FromState>
  <ToState>highLimit</ToState>
  <EventValues>
    <ChangeOfBitstring>
      <ReferencedBitstring>1</ReferencedBitstring>
      <StatusFlags>
        <inAlarm>true</inAlarm>
        <overridden>true</overridden>
      </StatusFlags>
    </ChangeOfBitstring>
  </EventValues>
</ConfirmedEventNotificationRequest>
```

Example 2:

```
<ConfirmedEventNotificationRequest>
  <ProcessIdentifier>2355</ProcessIdentifier>
  <InitiatingDeviceIdentifier>device-512</InitiatingDeviceIdentifier>
  <EventObjectIdentifier>device-513</EventObjectIdentifier>
  <Timestamp>
    <Time>22:39:15.980</Time>
  </Timestamp>
  <NotificationClass>1248</NotificationClass>
  <Priority>48</Priority>
  <EventType>changeOfReliability</EventType>
  <NotifyType>ackNotification</NotifyType>
  <ToState>normal</ToState>
</ConfirmedEventNotificationRequest>
```

Date

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.priorityArray**
- **analogOutput.priorityArray**

- **analogValue.priorityArray**
- **binaryLightingOutput.priorityArray**
- **binaryOutput.priorityArray**
- **binaryValue.priorityArray**
- **bitstringValue.priorityArray**
- **channel.presentValue**
- **characterstringValue.priorityArray**
- **dateValue.priorityArray**
- **datepatternValue.priorityArray**
- **datetimeValue.priorityArray**
- **datetimepatternValue.priorityArray**
- **integerValue.priorityArray**
- **largeAnalogValue.priorityArray**
- **lightingOutput.priorityArray**
- **multiStateOutput.priorityArray**
- **multiStateValue.priorityArray**
- **octetstringValue.priorityArray**
- **positiveIntegerValue.priorityArray**
- **timeValue.priorityArray**
- **timepatternValue.priorityArray**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="Date">
    <xs:restriction base="xs:string">
      <xs:pattern value="((19|20)\d\d|(\d\d|00))-(\d\d|00)-(\d\d|00)" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="Value" type="Date" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <Date>2017-12-20</Date>
</Value>
```

Double

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.priorityArray**
- **analogOutput.priorityArray**
- **analogValue.priorityArray**
- **binaryLightingOutput.priorityArray**
- **binaryOutput.priorityArray**
- **binaryValue.priorityArray**
- **bitstringValue.priorityArray**
- **channel.presentValue**
- **characterstringValue.priorityArray**
- **dateValue.priorityArray**
- **datepatternValue.priorityArray**
- **datetimeValue.priorityArray**
- **datetimepatternValue.priorityArray**
- **integerValue.priorityArray**
- **largeAnalogValue.priorityArray**
- **lightingOutput.priorityArray**
- **multiStateOutput.priorityArray**
- **multiStateValue.priorityArray**
- **octetstringValue.priorityArray**
- **positiveIntegerValue.priorityArray**
- **timeValue.priorityArray**
- **timepatternValue.priorityArray**
- **timer.stateChangeValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="Double">
    <xs:restriction base="xs:double" />
  </xs:simpleType>
  <xs:element name="Value" type="Double" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <Double>123456.789012</Double>
</Value>
```

Enumerated

This XML representation of complex data is used by the following objects/properties:

- **timer.stateChangeValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="Enumerated">
    <xs:restriction base="xs:integer" />
  </xs:simpleType>
  <xs:element name="Value" type="Enumerated" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <Enumerated>0</Enumerated>
</Value>
```

Error

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="Error">
  <xs:sequence>
    <xs:element name="ErrorClass" type="Error_ErrorClass" />
    <xs:element name="ErrorCode" type="Error_ErrorCode" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<Error>
  <ErrorClass>device</ErrorClass>
  <ErrorCode>configurationInProgress</ErrorCode>
</Error>
```

Error_ErrorClass

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema


```
<xs:simpleType name="Error_ErrorClass">
  <xs:restriction base="xs:string">
    <xs:enumeration value="device" />
    <xs:enumeration value="object" />
    <xs:enumeration value="property" />
    <xs:enumeration value="resources" />
    <xs:enumeration value="security" />
    <xs:enumeration value="services" />
    <xs:enumeration value="vt" />
    <xs:enumeration value="communication" />
  </xs:restriction>
</xs:simpleType>
```

Examples

Example 1:

```
<Error_ErrorClass>device</Error_ErrorClass>
```

Error_ErrorCode

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:simpleType name="Error_ErrorCode">
  <xs:restriction base="xs:string">
    <xs:enumeration value="other" />
    <xs:enumeration value="configurationInProgress" />
    <xs:enumeration value="deviceBusy" />
    <xs:enumeration value="dynamicCreationNotSupported" />
    <xs:enumeration value="fileAccessDenied" />
    <xs:enumeration value="inconsistentParameters" />
    <xs:enumeration value="inconsistentSelectionCriterion" />
    <xs:enumeration value="invalidDataType" />
    <xs:enumeration value="invalidFileAccessMethod" />
    <xs:enumeration value="invalidFileStartPosition" />
    <xs:enumeration value="invalidParameterDataType" />
    <xs:enumeration value="invalidTimestamp" />
    <xs:enumeration value="missingRequiredParameter" />
    <xs:enumeration value="noObjectsOfSpecifiedType" />
    <xs:enumeration value="noSpaceForObject" />
    <xs:enumeration value="noSpaceToAddListElement" />
    <xs:enumeration value="noSpaceToWriteProperty" />
    <xs:enumeration value="noVtSessionsAvailable" />
    <xs:enumeration value="propertyIsNotAList" />
    <xs:enumeration value="objectDeletionNotPermitted" />
    <xs:enumeration value="objectIdentifierAlreadyExists" />
    <xs:enumeration value="operationalProblem" />
    <xs:enumeration value="passwordFailure" />
    <xs:enumeration value="readAccessDenied" />
    <xs:enumeration value="serviceRequestDenied" />
    <xs:enumeration value="timeout" />
    <xs:enumeration value="unknownObject" />
    <xs:enumeration value="unknownProperty" />
    <xs:enumeration value="unknownVtClass" />
    <xs:enumeration value="unknownVtSession" />
    <xs:enumeration value="unsupportedObjectType" />
    <xs:enumeration value="valueOutOfRange" />
    <xs:enumeration value="vtSessionAlreadyClosed" />
    <xs:enumeration value="vtSessionTerminationFailure" />
    <xs:enumeration value="writeAccessDenied" />
    <xs:enumeration value="characterSetNotSupported" />
    <xs:enumeration value="invalidArrayIndex" />
    <xs:enumeration value="covSubscriptionFailed" />
    <xs:enumeration value="notCovProperty" />
    <xs:enumeration value="optionalFunctionalityNotSupported" />
    <xs:enumeration value="invalidConfigurationData" />
    <xs:enumeration value="datatypeNotSupported" />
    <xs:enumeration value="duplicateName" />
    <xs:enumeration value="duplicateObjectId" />
    <xs:enumeration value="propertyIsNotAnArray" />
    <xs:enumeration value="abortBufferOverflow" />
    <xs:enumeration value="abortInvalidApuInThisState" />
    <xs:enumeration value="abortPreemptedByHigherPriorityTask" />
    <xs:enumeration value="abortSegmentationNotSupported" />
    <xs:enumeration value="abortProprietary" />
    <xs:enumeration value="abortOther" />
    <xs:enumeration value="invalidTag" />
    <xs:enumeration value="networkDown" />
    <xs:enumeration value="rejectBufferOverflow" />
    <xs:enumeration value="rejectInconsistentParameters" />
    <xs:enumeration value="rejectInvalidParameterDataType" />
    <xs:enumeration value="rejectInvalidTag" />
    <xs:enumeration value="rejectMissingRequiredParameter" />
    <xs:enumeration value="rejectParameterOutOfRange" />
    <xs:enumeration value="rejectTooManyArguments" />
    <xs:enumeration value="rejectUndefinedEnumeration" />
    <xs:enumeration value="rejectUnrecognizedService" />
    <xs:enumeration value="rejectProprietary" />
    <xs:enumeration value="rejectOther" />
    <xs:enumeration value="unknownDevice" />
    <xs:enumeration value="unknownRoute" />
    <xs:enumeration value="valueNotInitialized" />
    <xs:enumeration value="invalidEventState" />
    <xs:enumeration value="noAlarmConfigured" />
  </xs:restriction>
</xs:simpleType>

```

```

<xs:enumeration value="logBufferFull" />
<xs:enumeration value="loggedValuePurged" />
<xs:enumeration value="noPropertySpecified" />
<xs:enumeration value="notConfiguredForTriggeredLogging" />
<xs:enumeration value="unknownSubscription" />
<xs:enumeration value="parameterOutOfRange" />
<xs:enumeration value="listElementNotFound" />
<xs:enumeration value="busy" />
<xs:enumeration value="communicationDisabled" />
<xs:enumeration value="success" />
<xs:enumeration value="accessDenied" />
<xs:enumeration value="badDestinationAddress" />
<xs:enumeration value="badDestinationDeviceId" />
<xs:enumeration value="badSignature" />
<xs:enumeration value="badSourceAddress" />
<xs:enumeration value="badTimestamp" />
<xs:enumeration value="cannotUseKey" />
<xs:enumeration value="cannotVerifyMessageId" />
<xs:enumeration value="correctKeyRevision" />
<xs:enumeration value="destinationDeviceIdRequired" />
<xs:enumeration value="duplicateMessage" />
<xs:enumeration value="encryptionNotConfigured" />
<xs:enumeration value="encryptionRequired" />
<xs:enumeration value="incorrectKey" />
<xs:enumeration value="invalidKeyData" />
<xs:enumeration value="keyUpdateInProgress" />
<xs:enumeration value="malformedMessage" />
<xs:enumeration value="notKeyServer" />
<xs:enumeration value="securityNotConfigured" />
<xs:enumeration value="sourceSecurityRequired" />
<xs:enumeration value="tooManyKeys" />
<xs:enumeration value="unknownAuthenticationType" />
<xs:enumeration value="unknownKey" />
<xs:enumeration value="unknownKeyRevision" />
<xs:enumeration value="unknownSourceMessage" />
<xs:enumeration value="notRouterToDnet" />
<xs:enumeration value="routerBusy" />
<xs:enumeration value="unknownNetworkMessage" />
<xs:enumeration value="messageTooLong" />
<xs:enumeration value="securityError" />
<xs:enumeration value="addressingError" />
<xs:enumeration value="writeBdtFailed" />
<xs:enumeration value="readBdtFailed" />
<xs:enumeration value="registerForeignDeviceFailed" />
<xs:enumeration value="readFdtFailed" />
<xs:enumeration value="deleteFdtEntryFailed" />
<xs:enumeration value="distributeBroadcastFailed" />
<xs:enumeration value="unknownFileSize" />
<xs:enumeration value="abortApduTooLong" />
<xs:enumeration value="abortApplicationExceededReplyTime" />
<xs:enumeration value="abortOutOfResources" />
<xs:enumeration value="abortTsmTimeout" />
<xs:enumeration value="abortWindowSizeOutOfRange" />
<xs:enumeration value="fileFull" />
<xs:enumeration value="inconsistentConfiguration" />
<xs:enumeration value="inconsistentObjectType" />
<xs:enumeration value="internalError" />
<xs:enumeration value="notConfigured" />
<xs:enumeration value="outOfMemory" />
<xs:enumeration value="valueTooLong" />
<xs:enumeration value="abortInsufficientSecurity" />
<xs:enumeration value="abortSecurityError" />
<xs:enumeration value="duplicateEntry" />
<xs:enumeration value="invalidValueInThisState" />
</xs:restriction>
</xs:simpleType>

```

Examples

Example 1:

```
<Error_ErrorCode>other</Error_ErrorCode>
```

INTEGER

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.priorityArray**
- **analogOutput.priorityArray**
- **analogValue.priorityArray**
- **binaryLightingOutput.priorityArray**
- **binaryOutput.priorityArray**
- **binaryValue.priorityArray**
- **bitstringValue.priorityArray**
- **channel.presentValue**
- **characterstringValue.priorityArray**
- **dateValue.priorityArray**
- **datepatternValue.priorityArray**
- **datetimeValue.priorityArray**
- **datetimepatternValue.priorityArray**
- **integerValue.priorityArray**
- **largeAnalogValue.priorityArray**
- **lightingOutput.priorityArray**
- **multiStateOutput.priorityArray**
- **multiStateValue.priorityArray**
- **octetstringValue.priorityArray**
- **positiveIntegerValue.priorityArray**
- **timeValue.priorityArray**
- **timepatternValue.priorityArray**
- **timer.stateChangeValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="INTEGER">
    <xs:restriction base="xs:integer" />
  </xs:simpleType>
  <xs:element name="Value" type="INTEGER" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <INTEGER>-800</INTEGER>
</Value>
```

Null

This XML representation of complex data is used by the following objects/properties:

- **timer.stateChangeValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="Null" />
  <xs:element name="Value" type="Null" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <Null />
</Value>
```

OctetString

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.priorityArray**
- **analogOutput.priorityArray**
- **analogValue.priorityArray**
- **binaryLightingOutput.priorityArray**
- **binaryOutput.priorityArray**
- **binaryValue.priorityArray**
- **bitstringValue.priorityArray**
- **channel.presentValue**
- **characterstringValue.priorityArray**
- **dateValue.priorityArray**
- **datepatternValue.priorityArray**
- **datetimeValue.priorityArray**
- **datetimepatternValue.priorityArray**
- **integerValue.priorityArray**
- **largeAnalogValue.priorityArray**

- **lightingOutput.priorityArray**
- **multiStateOutput.priorityArray**
- **multiStateValue.priorityArray**
- **octetstringValue.priorityArray**
- **positiveIntegerValue.priorityArray**
- **timeValue.priorityArray**
- **timepatternValue.priorityArray**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="OctetString">
    <xs:restriction base="xs:hexBinary" />
  </xs:simpleType>
  <xs:element name="Value" type="OctetString" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <OctetString>00010203040506070809</OctetString>
</Value>
```

REAL

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="REAL">
  <xs:restriction base="xs:double" />
</xs:simpleType>
```

Examples

Example 1:

```
<REAL>1234.567749</REAL>
```

ReadAccessResult

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="ReadAccessResult">
  <xs:sequence>
    <xs:element name="ObjectIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="ListOfResults" type="ReadAccessResult_ListOfResults" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<ReadAccessResult>
  <ObjectIdentifier>device-501</ObjectIdentifier>
  <ListOfResults />
</ReadAccessResult>
```

Example 2:

```
<ReadAccessResult>
  <ObjectIdentifier>device-502</ObjectIdentifier>
  <ListOfResults>
    <ListOfResults_Item>
      <PropertyIdentifier>action</PropertyIdentifier>
      <PropertyArrayIndex>1237</PropertyArrayIndex>
      <ReadResult>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
      </ReadResult>
    </ListOfResults_Item>
    <ListOfResults_Item>
      <PropertyIdentifier>activeText</PropertyIdentifier>
      <PropertyArrayIndex>1239</PropertyArrayIndex>
      <ReadResult>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
      </ReadResult>
    </ListOfResults_Item>
    <ListOfResults_Item>
      <PropertyIdentifier>alarmValue</PropertyIdentifier>
      <PropertyArrayIndex>1241</PropertyArrayIndex>
      <ReadResult>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
      </ReadResult>
    </ListOfResults_Item>
  </ListOfResults>
</ReadAccessResult>
```

ReadAccessResult_ListOfResults

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="ReadAccessResult_ListOfResults">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="ListOfResults_Item"
type="ReadAccessResult_ListOfResults_Item" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<SequenceOfReadAccessResult_ListOfResults />
```

Example 2:

```
<SequenceOfReadAccessResult_ListOfResults>
  <ListOfResults />
  <ListOfResults>
    <ListOfResults_Item>
      <PropertyIdentifier>ackedTransitions</PropertyIdentifier>
      <PropertyArrayIndex>1235</PropertyArrayIndex>
      <ReadResult>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
      </ReadResult>
    </ListOfResults_Item>
  </ListOfResults>
  <ListOfResults>
    <ListOfResults_Item>
      <PropertyIdentifier>action</PropertyIdentifier>
      <PropertyArrayIndex>1237</PropertyArrayIndex>
      <ReadResult>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
      </ReadResult>
    </ListOfResults_Item>
  </ListOfResults>
</SequenceOfReadAccessResult_ListOfResults>
```

Example 3:


```
<SequenceOfReadAccessResult_ListOfResults>
  <ListOfResults>
    <ListOfResults_Item>
      <PropertyIdentifier>activeText</PropertyIdentifier>
      <PropertyArrayIndex>1239</PropertyArrayIndex>
      <ReadResult>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
      </ReadResult>
    </ListOfResults_Item>
    <ListOfResults_Item>
      <PropertyIdentifier>alarmValue</PropertyIdentifier>
      <PropertyArrayIndex>1241</PropertyArrayIndex>
      <ReadResult>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
      </ReadResult>
    </ListOfResults_Item>
    <ListOfResults_Item>
      <PropertyIdentifier>all</PropertyIdentifier>
      <PropertyArrayIndex>1243</PropertyArrayIndex>
      <ReadResult>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
      </ReadResult>
    </ListOfResults_Item>
  </ListOfResults>
  <ListOfResults>
    <ListOfResults_Item>
      <PropertyIdentifier>apduSegmentTimeout</PropertyIdentifier>
      <PropertyArrayIndex>1245</PropertyArrayIndex>
      <ReadResult>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
      </ReadResult>
    </ListOfResults_Item>
  </ListOfResults>
  <ListOfResults>
    <ListOfResults_Item>
      <PropertyIdentifier>applicationSoftwareVersion</PropertyIdentifier>
      <PropertyArrayIndex>1247</PropertyArrayIndex>
      <ReadResult>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
      </ReadResult>
    </ListOfResults_Item>
  </ListOfResults>
</SequenceOfReadAccessResult_ListOfResults>
```

ReadAccessResult_ListOfResults_Item

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="ReadAccessResult_ListOfResults_Item">
  <xs:sequence>
    <xs:element name="PropertyIdentifier" type="BACnetPropertyIdentifier" />
    <xs:element minOccurs="0" name="PropertyArrayIndex" type="Unsigned" />
    <xs:element name="ReadResult" type="ReadAccessResult_ListOfResults_Item_ReadResult" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<ReadAccessResult_ListOfResults_Item>
  <PropertyIdentifier>ackedTransitions</PropertyIdentifier>
  <PropertyArrayIndex>1235</PropertyArrayIndex>
  <ReadResult>
    <PropertyValue>
      <Boolean>true</Boolean>
    </PropertyValue>
  </ReadResult>
</ReadAccessResult_ListOfResults_Item>
```

Example 2:

```
<ReadAccessResult_ListOfResults_Item>
  <PropertyIdentifier>action</PropertyIdentifier>
  <ReadResult>
    <PropertyValue>
      <Boolean>true</Boolean>
    </PropertyValue>
  </ReadResult>
</ReadAccessResult_ListOfResults_Item>
```

ReadAccessResult_ListOfResults_Item_ReadResult

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="ReadAccessResult_ListOfResults_Item_ReadResult">
  <xs:sequence>
    <xs:choice>
      <xs:element name="PropertyValue" type="Any" />
      <xs:element name="PropertyAccessError" type="Error" />
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<ReadAccessResult_ListOfResults_Item_ReadResult>
  <PropertyValue>
    <Boolean>true</Boolean>
  </PropertyValue>
</ReadAccessResult_ListOfResults_Item_ReadResult>
```

Example 2:

```
<ReadAccessResult_ListOfResults_Item_ReadResult>
  <PropertyAccessError>
    <ErrorClass>device</ErrorClass>
    <ErrorCode>configurationInProgress</ErrorCode>
  </PropertyAccessError>
</ReadAccessResult_ListOfResults_Item_ReadResult>
```

ReadAccessSpecification

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="ReadAccessSpecification">
  <xs:sequence>
    <xs:element name="ObjectIdentifier" type="BACnetObjectIdentifier" />
    <xs:element name="ListOfPropertyReferences" type="SequenceOfBACnetPropertyReference" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<ReadAccessSpecification>
  <ObjectIdentifier>device-501</ObjectIdentifier>
  <ListOfPropertyReferences />
</ReadAccessSpecification>
```

Example 2:

```
<ReadAccessSpecification>
  <ObjectIdentifier>device-502</ObjectIdentifier>
  <ListOfPropertyReferences>
    <BACnetPropertyReference>
      <PropertyIdentifier>action</PropertyIdentifier>
      <PropertyArrayIndex>1237</PropertyArrayIndex>
    </BACnetPropertyReference>
    <BACnetPropertyReference>
      <PropertyIdentifier>activeText</PropertyIdentifier>
      <PropertyArrayIndex>1239</PropertyArrayIndex>
    </BACnetPropertyReference>
    <BACnetPropertyReference>
      <PropertyIdentifier>alarmValue</PropertyIdentifier>
      <PropertyArrayIndex>1241</PropertyArrayIndex>
    </BACnetPropertyReference>
  </ListOfPropertyReferences>
</ReadAccessSpecification>
```

SequenceOfBACnetAccessCredentialDisableReason

This XML representation of complex data is used by the following objects/properties:

- **accessCredential.reasonForDisable**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetAccessCredentialDisableReason">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetAccessCredentialDisableReason"
        type="BACnetAccessCredentialDisableReason" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetAccessCredentialDisableReason" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAccessCredentialDisableReason />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAccessCredentialDisableReason>
    <BACnetAccessCredentialDisableReason>disabled</BACnetAccessCredentialDisableReason>
    <BACnetAccessCredentialDisableReason>disabledNeedsProvisioning</BACnetAccessCredentialDisableReason>
    <BACnetAccessCredentialDisableReason>disabledUnassigned</BACnetAccessCredentialDisableReason>
  </SequenceOfBACnetAccessCredentialDisableReason>
</Value>
```

SequenceOfBACnetAccessEvent

This XML representation of complex data is used by the following objects/properties:

- **accessPoint.accessAlarmEvents**
- **accessPoint.accessTransactionEvents**
- **accessPoint.failedAttemptEvents**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetAccessEvent">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetAccessEvent"
type="BACnetAccessEvent" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetAccessEvent" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAccessEvent />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAccessEvent>
    <BACnetAccessEvent>none</BACnetAccessEvent>
    <BACnetAccessEvent>granted</BACnetAccessEvent>
    <BACnetAccessEvent>muster</BACnetAccessEvent>
  </SequenceOfBACnetAccessEvent>
</Value>
```

SequenceOfBACnetAccessRule

This XML representation of complex data is used by the following objects/properties:

- **accessRights.negativeAccessRules**
- **accessRights.positiveAccessRules**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:msns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetAccessRule">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetAccessRule"
type="BACnetAccessRule" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetAccessRule" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAccessRule />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAccessRule>
    <BACnetAccessRule>
      <TimeRangeSpecifier>specified</TimeRangeSpecifier>
      <TimeRange>
        <ObjectIdentifier>device-502</ObjectIdentifier>
        <PropertyIdentifier>action</PropertyIdentifier>
        <PropertyArrayIndex>1237</PropertyArrayIndex>
        <DeviceIdentifier>device-505</DeviceIdentifier>
      </TimeRange>
      <LocationSpecifier>all</LocationSpecifier>
      <Location>
        <DeviceIdentifier>device-507</DeviceIdentifier>
        <ObjectIdentifier>device-508</ObjectIdentifier>
      </Location>
      <Enable>true</Enable>
    </BACnetAccessRule>
    <BACnetAccessRule>
      <TimeRangeSpecifier>specified</TimeRangeSpecifier>
      <TimeRange>
        <ObjectIdentifier>device-510</ObjectIdentifier>
        <PropertyIdentifier>apduSegmentTimeout</PropertyIdentifier>
        <PropertyArrayIndex>1245</PropertyArrayIndex>
        <DeviceIdentifier>device-513</DeviceIdentifier>
      </TimeRange>
      <LocationSpecifier>all</LocationSpecifier>
      <Location>
        <DeviceIdentifier>device-515</DeviceIdentifier>
        <ObjectIdentifier>device-516</ObjectIdentifier>
      </Location>
      <Enable>true</Enable>
    </BACnetAccessRule>
    <BACnetAccessRule>
      <TimeRangeSpecifier>specified</TimeRangeSpecifier>
      <TimeRange>
        <ObjectIdentifier>device-518</ObjectIdentifier>
        <PropertyIdentifier>controlledVariableReference</PropertyIdentifier>
        <PropertyArrayIndex>1253</PropertyArrayIndex>
        <DeviceIdentifier>device-521</DeviceIdentifier>
      </TimeRange>
      <LocationSpecifier>all</LocationSpecifier>
      <Location>
        <DeviceIdentifier>device-523</DeviceIdentifier>
        <ObjectIdentifier>device-524</ObjectIdentifier>
      </Location>
      <Enable>true</Enable>
    </BACnetAccessRule>
  </SequenceOfBACnetAccessRule>
</Value>
```

SequenceOfBACnetAccessZoneOccupancyState

This XML representation of complex data is used by the following objects/properties:

- **accessZone.alarmValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetAccessZoneOccupancyState">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetAccessZoneOccupancyState"
type="BACnetAccessZoneOccupancyState" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetAccessZoneOccupancyState" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAccessZoneOccupancyState />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAccessZoneOccupancyState>
    <BACnetAccessZoneOccupancyState>normal</BACnetAccessZoneOccupancyState>
    <BACnetAccessZoneOccupancyState>belowLowerLimit</BACnetAccessZoneOccupancyState>
    <BACnetAccessZoneOccupancyState>atLowerLimit</BACnetAccessZoneOccupancyState>
  </SequenceOfBACnetAccessZoneOccupancyState>
</Value>
```

SequenceOfBACnetActionCommand

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="SequenceOfBACnetActionCommand">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetActionCommand"
type="BACnetActionCommand" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<SequenceOfBACnetActionCommand />
```

Example 2:

```

<SequenceOfBACnetActionCommand>
  <BACnetActionCommand>
    <DeviceIdentifier>device-501</DeviceIdentifier>
    <ObjectIdentifier>device-502</ObjectIdentifier>
    <PropertyIdentifier>action</PropertyIdentifier>
    <PropertyArrayIndex>1237</PropertyArrayIndex>
    <PropertyValue>
      <Boolean>true</Boolean>
    </PropertyValue>
    <Priority>1238</Priority>
    <PostDelay>1239</PostDelay>
    <QuitOnFailure>true</QuitOnFailure>
    <WriteSuccessful>true</WriteSuccessful>
  </BACnetActionCommand>
  <BACnetActionCommand>
    <DeviceIdentifier>device-507</DeviceIdentifier>
    <ObjectIdentifier>device-508</ObjectIdentifier>
    <PropertyIdentifier>all</PropertyIdentifier>
    <PropertyArrayIndex>1243</PropertyArrayIndex>
    <PropertyValue>
      <Boolean>true</Boolean>
    </PropertyValue>
    <Priority>1244</Priority>
    <PostDelay>1245</PostDelay>
    <QuitOnFailure>true</QuitOnFailure>
    <WriteSuccessful>true</WriteSuccessful>
  </BACnetActionCommand>
  <BACnetActionCommand>
    <DeviceIdentifier>device-513</DeviceIdentifier>
    <ObjectIdentifier>device-514</ObjectIdentifier>
    <PropertyIdentifier>bias</PropertyIdentifier>
    <PropertyArrayIndex>1249</PropertyArrayIndex>
    <PropertyValue>
      <Boolean>true</Boolean>
    </PropertyValue>
    <Priority>1250</Priority>
    <PostDelay>1251</PostDelay>
    <QuitOnFailure>true</QuitOnFailure>
    <WriteSuccessful>true</WriteSuccessful>
  </BACnetActionCommand>
</SequenceOfBACnetActionCommand>

```

SequenceOfBACnetActionList

This XML representation of complex data is used by the following objects/properties:

- **command.action**

XML Schema

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetActionList">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetActionList"
type="BACnetActionList" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetActionList" />
</xs:schema>

```

Examples

Example 1:


```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetActionList />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetActionList>
    <BACnetActionList>
      <Action />
    </BACnetActionList>
    <BACnetActionList>
      <Action>
        <BACnetActionCommand>
          <DeviceIdentifier>device-501</DeviceIdentifier>
          <ObjectIdentifier>device-502</ObjectIdentifier>
          <PropertyIdentifier>action</PropertyIdentifier>
          <PropertyArrayIndex>1237</PropertyArrayIndex>
          <PropertyValue>
            <Boolean>true</Boolean>
          </PropertyValue>
          <Priority>1238</Priority>
          <PostDelay>1239</PostDelay>
          <QuitOnFailure>true</QuitOnFailure>
          <WriteSuccessful>true</WriteSuccessful>
        </BACnetActionCommand>
      </Action>
    </BACnetActionList>
    <BACnetActionList>
      <Action>
        <BACnetActionCommand>
          <DeviceIdentifier>device-507</DeviceIdentifier>
          <ObjectIdentifier>device-508</ObjectIdentifier>
          <PropertyIdentifier>all</PropertyIdentifier>
          <PropertyArrayIndex>1243</PropertyArrayIndex>
          <PropertyValue>
            <Boolean>true</Boolean>
          </PropertyValue>
          <Priority>1244</Priority>
          <PostDelay>1245</PostDelay>
          <QuitOnFailure>true</QuitOnFailure>
          <WriteSuccessful>true</WriteSuccessful>
        </BACnetActionCommand>
      </Action>
    </BACnetActionList>
  </SequenceOfBACnetActionList>
</Value>
```

Example 3:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetActionList>
    <BACnetActionList>
      <Action>
        <BACnetActionCommand>
          <DeviceIdentifier>device-513</DeviceIdentifier>
          <ObjectIdentifier>device-514</ObjectIdentifier>
          <PropertyIdentifier>bias</PropertyIdentifier>
          <PropertyArrayIndex>1249</PropertyArrayIndex>
          <PropertyValue>
            <Boolean>true</Boolean>
          </PropertyValue>
          <Priority>1250</Priority>
          <PostDelay>1251</PostDelay>
          <QuitOnFailure>true</QuitOnFailure>
          <WriteSuccessful>true</WriteSuccessful>
        </BACnetActionCommand>
      </BACnetActionCommand>
      <BACnetActionCommand>
        <DeviceIdentifier>device-519</DeviceIdentifier>
        <ObjectIdentifier>device-520</ObjectIdentifier>
        <PropertyIdentifier>controlledVariableValue</PropertyIdentifier>
        <PropertyArrayIndex>1255</PropertyArrayIndex>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
        <Priority>1256</Priority>
        <PostDelay>1257</PostDelay>
        <QuitOnFailure>true</QuitOnFailure>
        <WriteSuccessful>true</WriteSuccessful>
      </BACnetActionCommand>
      <BACnetActionCommand>
        <DeviceIdentifier>device-525</DeviceIdentifier>
        <ObjectIdentifier>device-526</ObjectIdentifier>
        <PropertyIdentifier>derivativeConstantUnits</PropertyIdentifier>
        <PropertyArrayIndex>1261</PropertyArrayIndex>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
        <Priority>1262</Priority>
        <PostDelay>1263</PostDelay>
        <QuitOnFailure>true</QuitOnFailure>
        <WriteSuccessful>true</WriteSuccessful>
      </BACnetActionCommand>
    </Action>
  </BACnetActionList>
  <BACnetActionList>
    <Action>
      <BACnetActionCommand>
        <DeviceIdentifier>device-531</DeviceIdentifier>
        <ObjectIdentifier>device-532</ObjectIdentifier>
        <PropertyIdentifier>elapsedActiveTime</PropertyIdentifier>
        <PropertyArrayIndex>1267</PropertyArrayIndex>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
        <Priority>1268</Priority>
        <PostDelay>1269</PostDelay>
        <QuitOnFailure>true</QuitOnFailure>
        <WriteSuccessful>true</WriteSuccessful>
      </BACnetActionCommand>
    </Action>
  </BACnetActionList>
  <BACnetActionList>
    <Action>
      <BACnetActionCommand>
        <DeviceIdentifier>device-537</DeviceIdentifier>
        <ObjectIdentifier>device-538</ObjectIdentifier>
        <PropertyIdentifier>faultValues</PropertyIdentifier>
        <PropertyArrayIndex>1273</PropertyArrayIndex>
        <PropertyValue>
```

```

        <Boolean>true</Boolean>
      </PropertyValue>
      <Priority>1274</Priority>
      <PostDelay>1275</PostDelay>
      <QuitOnFailure>true</QuitOnFailure>
      <WriteSuccessful>true</WriteSuccessful>
    </BACnetActionCommand>
  </Action>
</BACnetActionList>
</SequenceOfBACnetActionList>
</Value>

```

SequenceOfBACnetAddressBinding

This XML representation of complex data is used by the following objects/properties:

- **device.deviceAddressBinding**
- **networkPort.manualSlaveAddressBinding**
- **networkPort.slaveAddressBinding**

XML Schema

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetAddressBinding">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetAddressBinding"
type="BACnetAddressBinding" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetAddressBinding" />
</xs:schema>

```

Examples

Example 1:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAddressBinding />
</Value>

```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAddressBinding>
    <BACnetAddressBinding>
      <DeviceIdentifier>device-501</DeviceIdentifier>
      <DeviceAddress>
        <NetworkNumber>7635</NetworkNumber>
        <MacAddress>00010203040506070809</MacAddress>
      </DeviceAddress>
    </BACnetAddressBinding>
    <BACnetAddressBinding>
      <DeviceIdentifier>device-503</DeviceIdentifier>
      <DeviceAddress>
        <NetworkNumber>7637</NetworkNumber>
        <MacAddress>FF</MacAddress>
      </DeviceAddress>
    </BACnetAddressBinding>
    <BACnetAddressBinding>
      <DeviceIdentifier>device-505</DeviceIdentifier>
      <DeviceAddress>
        <NetworkNumber>7639</NetworkNumber>
        <MacAddress>FF</MacAddress>
      </DeviceAddress>
    </BACnetAddressBinding>
  </SequenceOfBACnetAddressBinding>
</Value>
```

SequenceOfBACnetAssignedAccessRights

This XML representation of complex data is used by the following objects/properties:

- **accessCredential.assignedAccessRights**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetAssignedAccessRights">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetAssignedAccessRights"
type="BACnetAssignedAccessRights" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetAssignedAccessRights" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAssignedAccessRights />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAssignedAccessRights>
    <BACnetAssignedAccessRights>
      <AssignedAccessRights>
        <DeviceIdentifier>device-501</DeviceIdentifier>
        <ObjectIdentifier>device-502</ObjectIdentifier>
      </AssignedAccessRights>
      <Enable>true</Enable>
    </BACnetAssignedAccessRights>
    <BACnetAssignedAccessRights>
      <AssignedAccessRights>
        <DeviceIdentifier>device-503</DeviceIdentifier>
        <ObjectIdentifier>device-504</ObjectIdentifier>
      </AssignedAccessRights>
      <Enable>true</Enable>
    </BACnetAssignedAccessRights>
    <BACnetAssignedAccessRights>
      <AssignedAccessRights>
        <DeviceIdentifier>device-505</DeviceIdentifier>
        <ObjectIdentifier>device-506</ObjectIdentifier>
      </AssignedAccessRights>
      <Enable>true</Enable>
    </BACnetAssignedAccessRights>
  </SequenceOfBACnetAssignedAccessRights>
</Value>
```

SequenceOfBACnetAssignedLandingCalls

This XML representation of complex data is used by the following objects/properties:

- **lift.assignedLandingCalls**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetAssignedLandingCalls">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetAssignedLandingCalls"
type="BACnetAssignedLandingCalls" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetAssignedLandingCalls" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAssignedLandingCalls />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAssignedLandingCalls>
    <BACnetAssignedLandingCalls>
      <LandingCalls />
    </BACnetAssignedLandingCalls>
    <BACnetAssignedLandingCalls>
      <LandingCalls>
        <LandingCalls_Item>
          <FloorNumber>33</FloorNumber>
          <Direction>none</Direction>
        </LandingCalls_Item>
      </LandingCalls>
    </BACnetAssignedLandingCalls>
    <BACnetAssignedLandingCalls>
      <LandingCalls>
        <LandingCalls_Item>
          <FloorNumber>35</FloorNumber>
          <Direction>up</Direction>
        </LandingCalls_Item>
      </LandingCalls>
    </BACnetAssignedLandingCalls>
  </SequenceOfBACnetAssignedLandingCalls>
</Value>
```

Example 3:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAssignedLandingCalls>
    <BACnetAssignedLandingCalls>
      <LandingCalls>
        <LandingCalls_Item>
          <FloorNumber>37</FloorNumber>
          <Direction>upAndDown</Direction>
        </LandingCalls_Item>
        <LandingCalls_Item>
          <FloorNumber>39</FloorNumber>
          <Direction>none</Direction>
        </LandingCalls_Item>
        <LandingCalls_Item>
          <FloorNumber>41</FloorNumber>
          <Direction>up</Direction>
        </LandingCalls_Item>
      </LandingCalls>
    </BACnetAssignedLandingCalls>
    <BACnetAssignedLandingCalls>
      <LandingCalls>
        <LandingCalls_Item>
          <FloorNumber>43</FloorNumber>
          <Direction>upAndDown</Direction>
        </LandingCalls_Item>
      </LandingCalls>
    </BACnetAssignedLandingCalls>
    <BACnetAssignedLandingCalls>
      <LandingCalls>
        <LandingCalls_Item>
          <FloorNumber>45</FloorNumber>
          <Direction>none</Direction>
        </LandingCalls_Item>
      </LandingCalls>
    </BACnetAssignedLandingCalls>
  </SequenceOfBACnetAssignedLandingCalls>
</Value>
```

SequenceOfBACnetAuthenticationFactorFormat

This XML representation of complex data is used by the following objects/properties:

- **credentialDataInput.supportedFormats**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetAuthenticationFactorFormat">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetAuthenticationFactorFormat"
type="BACnetAuthenticationFactorFormat" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetAuthenticationFactorFormat" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAuthenticationFactorFormat />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAuthenticationFactorFormat>
    <BACnetAuthenticationFactorFormat>
      <FormatType>undefined</FormatType>
      <VendorId>7635</VendorId>
      <VendorFormat>7636</VendorFormat>
    </BACnetAuthenticationFactorFormat>
    <BACnetAuthenticationFactorFormat>
      <FormatType>simpleNumber16</FormatType>
      <VendorId>7638</VendorId>
      <VendorFormat>7639</VendorFormat>
    </BACnetAuthenticationFactorFormat>
    <BACnetAuthenticationFactorFormat>
      <FormatType>simpleAlphaNumeric</FormatType>
      <VendorId>7641</VendorId>
      <VendorFormat>7642</VendorFormat>
    </BACnetAuthenticationFactorFormat>
  </SequenceOfBACnetAuthenticationFactorFormat>
</Value>
```

SequenceOfBACnetAuthenticationPolicy

This XML representation of complex data is used by the following objects/properties:

- **accessPoint.authenticationPolicyList**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetAuthenticationPolicy">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetAuthenticationPolicy"
type="BACnetAuthenticationPolicy" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetAuthenticationPolicy" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAuthenticationPolicy />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAuthenticationPolicy>
    <BACnetAuthenticationPolicy>
      <Policy />
      <OrderEnforced>true</OrderEnforced>
      <Timeout>1234</Timeout>
    </BACnetAuthenticationPolicy>
    <BACnetAuthenticationPolicy>
      <Policy>
        <Policy_Item>
          <CredentialDataInput>
            <DeviceIdentifier>device-502</DeviceIdentifier>
            <ObjectIdentifier>device-503</ObjectIdentifier>
          </CredentialDataInput>
          <Index>1237</Index>
        </Policy_Item>
      </Policy>
      <OrderEnforced>true</OrderEnforced>
      <Timeout>1238</Timeout>
    </BACnetAuthenticationPolicy>
    <BACnetAuthenticationPolicy>
      <Policy>
        <Policy_Item>
          <CredentialDataInput>
            <DeviceIdentifier>device-506</DeviceIdentifier>
            <ObjectIdentifier>device-507</ObjectIdentifier>
          </CredentialDataInput>
          <Index>1241</Index>
        </Policy_Item>
      </Policy>
      <OrderEnforced>true</OrderEnforced>
      <Timeout>1242</Timeout>
    </BACnetAuthenticationPolicy>
  </SequenceOfBACnetAuthenticationPolicy>
</Value>
```

Example 3:


```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAuthenticationPolicy>
    <BACnetAuthenticationPolicy>
      <Policy>
        <Policy_Item>
          <CredentialDataInput>
            <DeviceIdentifier>device-510</DeviceIdentifier>
            <ObjectIdentifier>device-511</ObjectIdentifier>
          </CredentialDataInput>
          <Index>1245</Index>
        </Policy_Item>
        <Policy_Item>
          <CredentialDataInput>
            <DeviceIdentifier>device-513</DeviceIdentifier>
            <ObjectIdentifier>device-514</ObjectIdentifier>
          </CredentialDataInput>
          <Index>1248</Index>
        </Policy_Item>
        <Policy_Item>
          <CredentialDataInput>
            <DeviceIdentifier>device-516</DeviceIdentifier>
            <ObjectIdentifier>device-517</ObjectIdentifier>
          </CredentialDataInput>
          <Index>1251</Index>
        </Policy_Item>
      </Policy>
      <OrderEnforced>true</OrderEnforced>
      <Timeout>1252</Timeout>
    </BACnetAuthenticationPolicy>
    <BACnetAuthenticationPolicy>
      <Policy>
        <Policy_Item>
          <CredentialDataInput>
            <DeviceIdentifier>device-520</DeviceIdentifier>
            <ObjectIdentifier>device-521</ObjectIdentifier>
          </CredentialDataInput>
          <Index>1255</Index>
        </Policy_Item>
      </Policy>
      <OrderEnforced>true</OrderEnforced>
      <Timeout>1256</Timeout>
    </BACnetAuthenticationPolicy>
    <BACnetAuthenticationPolicy>
      <Policy>
        <Policy_Item>
          <CredentialDataInput>
            <DeviceIdentifier>device-524</DeviceIdentifier>
            <ObjectIdentifier>device-525</ObjectIdentifier>
          </CredentialDataInput>
          <Index>1259</Index>
        </Policy_Item>
      </Policy>
      <OrderEnforced>true</OrderEnforced>
      <Timeout>1260</Timeout>
    </BACnetAuthenticationPolicy>
  </SequenceOfBACnetAuthenticationPolicy>
</Value>

```

SequenceOfBACnetAuthorizationExemption

This XML representation of complex data is used by the following objects/properties:

- **accessCredential.authorizationExemptions**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetAuthorizationExemption">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetAuthorizationExemption"
type="BACnetAuthorizationExemption" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetAuthorizationExemption" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAuthorizationExemption />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetAuthorizationExemption>
    <BACnetAuthorizationExemption>passback</BACnetAuthorizationExemption>
    <BACnetAuthorizationExemption>occupancyCheck</BACnetAuthorizationExemption>
    <BACnetAuthorizationExemption>accessRights</BACnetAuthorizationExemption>
  </SequenceOfBACnetAuthorizationExemption>
</Value>
```

SequenceOfBACnetBDTEntry

This XML representation of complex data is used by the following objects/properties:

- **networkPort.bbmdBroadcastDistributionTable**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetBDTEntry">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetBDTEntry" type="BACnetBDTEntry" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetBDTEntry" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetBDTEntry />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetBDTEntry>
    <BACnetBDTEntry>
      <BbmdAddress>
        <Host>
          <None />
        </Host>
        <Port>7634</Port>
      </BbmdAddress>
      <BroadcastMask>00010203040506070809</BroadcastMask>
    </BACnetBDTEntry>
    <BACnetBDTEntry>
      <BbmdAddress>
        <Host>
          <None />
        </Host>
        <Port>7635</Port>
      </BbmdAddress>
      <BroadcastMask>FF</BroadcastMask>
    </BACnetBDTEntry>
    <BACnetBDTEntry>
      <BbmdAddress>
        <Host>
          <None />
        </Host>
        <Port>7636</Port>
      </BbmdAddress>
      <BroadcastMask>FF</BroadcastMask>
    </BACnetBDTEntry>
  </SequenceOfBACnetBDTEntry>
</Value>
```

SequenceOfBACnetCOVMultipleSubscription

This XML representation of complex data is used by the following objects/properties:

- **device.activeCovMultipleSubscriptions**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:msns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetCOVMultipleSubscription">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetCOVMultipleSubscription"
type="BACnetCOVMultipleSubscription" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetCOVMultipleSubscription" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetCOVMultipleSubscription />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetCOVMultipleSubscription>
    <BACnetCOVMultipleSubscription>
      <Recipient>
        <Recipient>
          <Device>device-501</Device>
        </Recipient>
        <ProcessIdentifier>2346</ProcessIdentifier>
      </Recipient>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <TimeRemaining>1236</TimeRemaining>
      <MaxNotificationDelay>1237</MaxNotificationDelay>
      <ListOfCovSubscriptionSpecifications />
    </BACnetCOVMultipleSubscription>
    <BACnetCOVMultipleSubscription>
      <Recipient>
        <Recipient>
          <Device>device-505</Device>
        </Recipient>
        <ProcessIdentifier>2350</ProcessIdentifier>
      </Recipient>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <TimeRemaining>1240</TimeRemaining>
      <MaxNotificationDelay>1241</MaxNotificationDelay>
      <ListOfCovSubscriptionSpecifications>
        <ListOfCovSubscriptionSpecifications_Item>
          <MonitoredObjectIdentifier>device-509</MonitoredObjectIdentifier>
          <ListOfCovReferences>
            <ListOfCovReferences_Item>
              <MonitoredProperty>
                <PropertyIdentifier>allWritesSuccessful</PropertyIdentifier>
                <PropertyArrayIndex>1244</PropertyArrayIndex>
              </MonitoredProperty>
              <CovIncrement>1245.567749</CovIncrement>
              <Timestamped>true</Timestamped>
            </ListOfCovReferences_Item>
          </ListOfCovReferences>
        </ListOfCovSubscriptionSpecifications_Item>
      </ListOfCovSubscriptionSpecifications>
    </BACnetCOVMultipleSubscription>
    <BACnetCOVMultipleSubscription>
      <Recipient>
        <Recipient>
          <Device>device-513</Device>
        </Recipient>
        <ProcessIdentifier>2358</ProcessIdentifier>
      </Recipient>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <TimeRemaining>1248</TimeRemaining>
      <MaxNotificationDelay>1249</MaxNotificationDelay>
      <ListOfCovSubscriptionSpecifications>
        <ListOfCovSubscriptionSpecifications_Item>
          <MonitoredObjectIdentifier>device-517</MonitoredObjectIdentifier>
          <ListOfCovReferences>
            <ListOfCovReferences_Item>
              <MonitoredProperty>
                <PropertyIdentifier>notificationClass</PropertyIdentifier>
                <PropertyArrayIndex>1252</PropertyArrayIndex>
              </MonitoredProperty>
              <CovIncrement>1253.567749</CovIncrement>
              <Timestamped>true</Timestamped>
            </ListOfCovReferences_Item>
          </ListOfCovReferences>
        </ListOfCovSubscriptionSpecifications_Item>
      </ListOfCovSubscriptionSpecifications>
    </BACnetCOVMultipleSubscription>
  </SequenceOfBACnetCOVMultipleSubscription>
</Value>
```

Example 3:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetCOVMultipleSubscription>
    <BACnetCOVMultipleSubscription>
      <Recipient>
        <Recipient>
          <Device>device-521</Device>
        </Recipient>
        <ProcessIdentifier>2366</ProcessIdentifier>
      </Recipient>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <TimeRemaining>1256</TimeRemaining>
      <MaxNotificationDelay>1257</MaxNotificationDelay>
      <ListOfCovSubscriptionSpecifications>
        <ListOfCovSubscriptionSpecifications_Item>
          <MonitoredObjectIdentifier>device-525</MonitoredObjectIdentifier>
          <ListOfCovReferences />
        </ListOfCovSubscriptionSpecifications_Item>
        <ListOfCovSubscriptionSpecifications_Item>
          <MonitoredObjectIdentifier>device-526</MonitoredObjectIdentifier>
          <ListOfCovReferences>
            <ListOfCovReferences_Item>
              <MonitoredProperty>
                <PropertyIdentifier>derivativeConstantUnits</PropertyIdentifier>
                <PropertyArrayIndex>1261</PropertyArrayIndex>
              </MonitoredProperty>
              <CovIncrement>1262.567749</CovIncrement>
              <Timestamped>true</Timestamped>
            </ListOfCovReferences_Item>
          </ListOfCovReferences>
        </ListOfCovSubscriptionSpecifications_Item>
        <ListOfCovSubscriptionSpecifications_Item>
          <MonitoredObjectIdentifier>device-530</MonitoredObjectIdentifier>
          <ListOfCovReferences>
            <ListOfCovReferences_Item>
              <MonitoredProperty>
                <PropertyIdentifier>deviceType</PropertyIdentifier>
                <PropertyArrayIndex>1265</PropertyArrayIndex>
              </MonitoredProperty>
              <CovIncrement>1266.567749</CovIncrement>
              <Timestamped>true</Timestamped>
            </ListOfCovReferences_Item>
          </ListOfCovReferences>
        </ListOfCovSubscriptionSpecifications_Item>
      </ListOfCovSubscriptionSpecifications>
    </BACnetCOVMultipleSubscription>
    <BACnetCOVMultipleSubscription>
      <Recipient>
        <Recipient>
          <Device>device-534</Device>
        </Recipient>
        <ProcessIdentifier>2379</ProcessIdentifier>
      </Recipient>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <TimeRemaining>1269</TimeRemaining>
      <MaxNotificationDelay>1270</MaxNotificationDelay>
      <ListOfCovSubscriptionSpecifications>
        <ListOfCovSubscriptionSpecifications_Item>
          <MonitoredObjectIdentifier>device-538</MonitoredObjectIdentifier>
          <ListOfCovReferences>
            <ListOfCovReferences_Item>
              <MonitoredProperty>
                <PropertyIdentifier>faultValues</PropertyIdentifier>
                <PropertyArrayIndex>1273</PropertyArrayIndex>
              </MonitoredProperty>
              <CovIncrement>1274.567749</CovIncrement>
              <Timestamped>true</Timestamped>
            </ListOfCovReferences_Item>
          </ListOfCovReferences>
        </ListOfCovSubscriptionSpecifications_Item>
      </ListOfCovSubscriptionSpecifications>
    </BACnetCOVMultipleSubscription>
  </SequenceOfBACnetCOVMultipleSubscription>
</Value>
```

```

</BACnetCOVMultipleSubscription>
<BACnetCOVMultipleSubscription>
  <Recipient>
    <Recipient>
      <Device>device-542</Device>
    </Recipient>
    <ProcessIdentifier>2387</ProcessIdentifier>
  </Recipient>
  <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
  <TimeRemaining>1277</TimeRemaining>
  <MaxNotificationDelay>1278</MaxNotificationDelay>
  <ListOfCovSubscriptionSpecifications>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-546</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>inProcess</PropertyIdentifier>
            <PropertyArrayIndex>1281</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1282.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>
</BACnetCOVMultipleSubscription>
</SequenceOfBACnetCOVMultipleSubscription>
</Value>

```

Example 4:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetCOVMultipleSubscription>
    <BACnetCOVMultipleSubscription>
      <Recipient>
        <Recipient>
          <Device>device-550</Device>
        </Recipient>
        <ProcessIdentifier>2395</ProcessIdentifier>
      </Recipient>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <TimeRemaining>1285</TimeRemaining>
      <MaxNotificationDelay>1286</MaxNotificationDelay>
      <ListOfCovSubscriptionSpecifications>
        <ListOfCovSubscriptionSpecifications_Item>
          <MonitoredObjectIdentifier>device-554</MonitoredObjectIdentifier>
          <ListOfCovReferences>
            <ListOfCovReferences_Item>
              <MonitoredProperty>
                <PropertyIdentifier>localTime</PropertyIdentifier>
                <PropertyArrayIndex>1289</PropertyArrayIndex>
              </MonitoredProperty>
              <CovIncrement>1290.567749</CovIncrement>
              <Timestamped>true</Timestamped>
            </ListOfCovReferences_Item>
            <ListOfCovReferences_Item>
              <MonitoredProperty>
                <PropertyIdentifier>manipulatedVariableReference</PropertyIdentifier>
                <PropertyArrayIndex>1292</PropertyArrayIndex>
              </MonitoredProperty>
              <CovIncrement>1293.567749</CovIncrement>
              <Timestamped>true</Timestamped>
            </ListOfCovReferences_Item>
            <ListOfCovReferences_Item>
              <MonitoredProperty>
                <PropertyIdentifier>maxInfoFrames</PropertyIdentifier>
                <PropertyArrayIndex>1295</PropertyArrayIndex>
              </MonitoredProperty>
              <CovIncrement>1296.567749</CovIncrement>
              <Timestamped>true</Timestamped>
            </ListOfCovReferences_Item>
          </ListOfCovReferences>
        </ListOfCovSubscriptionSpecifications_Item>
        <ListOfCovSubscriptionSpecifications_Item>
          <MonitoredObjectIdentifier>device-564</MonitoredObjectIdentifier>
          <ListOfCovReferences>
            <ListOfCovReferences_Item>
              <MonitoredProperty>
                <PropertyIdentifier>minimumOnTime</PropertyIdentifier>
                <PropertyArrayIndex>1299</PropertyArrayIndex>
              </MonitoredProperty>
              <CovIncrement>1300.567749</CovIncrement>
              <Timestamped>true</Timestamped>
            </ListOfCovReferences_Item>
          </ListOfCovReferences>
        </ListOfCovSubscriptionSpecifications_Item>
        <ListOfCovSubscriptionSpecifications_Item>
          <MonitoredObjectIdentifier>device-568</MonitoredObjectIdentifier>
          <ListOfCovReferences>
            <ListOfCovReferences_Item>
              <MonitoredProperty>
                <PropertyIdentifier>modificationDate</PropertyIdentifier>
                <PropertyArrayIndex>1303</PropertyArrayIndex>
              </MonitoredProperty>
              <CovIncrement>1304.567749</CovIncrement>
              <Timestamped>true</Timestamped>
            </ListOfCovReferences_Item>
          </ListOfCovReferences>
        </ListOfCovSubscriptionSpecifications_Item>
      </ListOfCovSubscriptionSpecifications>
    </BACnetCOVMultipleSubscription>
  </SequenceOfBACnetCOVMultipleSubscription>
</Value>
```

```

<BACnetCOVMultipleSubscription>
  <Recipient>
    <Recipient>
      <Device>device-572</Device>
    </Recipient>
    <ProcessIdentifier>2417</ProcessIdentifier>
  </Recipient>
  <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
  <TimeRemaining>1307</TimeRemaining>
  <MaxNotificationDelay>1308</MaxNotificationDelay>
  <ListOfCovSubscriptionSpecifications>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-576</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>objectType</PropertyIdentifier>
            <PropertyArrayIndex>1311</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1312.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>
</BACnetCOVMultipleSubscription>
<BACnetCOVMultipleSubscription>
  <Recipient>
    <Recipient>
      <Device>device-580</Device>
    </Recipient>
    <ProcessIdentifier>2425</ProcessIdentifier>
  </Recipient>
  <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
  <TimeRemaining>1315</TimeRemaining>
  <MaxNotificationDelay>1316</MaxNotificationDelay>
  <ListOfCovSubscriptionSpecifications>
    <ListOfCovSubscriptionSpecifications_Item>
      <MonitoredObjectIdentifier>device-584</MonitoredObjectIdentifier>
      <ListOfCovReferences>
        <ListOfCovReferences_Item>
          <MonitoredProperty>
            <PropertyIdentifier>priorityArray</PropertyIdentifier>
            <PropertyArrayIndex>1319</PropertyArrayIndex>
          </MonitoredProperty>
          <CovIncrement>1320.567749</CovIncrement>
          <Timestamped>true</Timestamped>
        </ListOfCovReferences_Item>
      </ListOfCovReferences>
    </ListOfCovSubscriptionSpecifications_Item>
  </ListOfCovSubscriptionSpecifications>
</BACnetCOVMultipleSubscription>
</SequenceOfBACnetCOVMultipleSubscription>
</Value>

```

SequenceOfBACnetCOVSubscription

This XML representation of complex data is used by the following objects/properties:

- **device.activeCovSubscriptions**

XML Schema


```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:msns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetCOVSubscription">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetCOVSubscription"
type="BACnetCOVSubscription" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetCOVSubscription" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetCOVSubscription />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetCOVSubscription>
    <BACnetCOVSubscription>
      <Recipient>
        <Recipient>
          <Device>device-501</Device>
        </Recipient>
        <ProcessIdentifier>2346</ProcessIdentifier>
      </Recipient>
      <MonitoredPropertyReference>
        <ObjectIdentifier>device-503</ObjectIdentifier>
        <PropertyIdentifier>actionText</PropertyIdentifier>
        <PropertyArrayIndex>1238</PropertyArrayIndex>
      </MonitoredPropertyReference>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <TimeRemaining>1239</TimeRemaining>
      <CovIncrement>1240.567749</CovIncrement>
    </BACnetCOVSubscription>
    <BACnetCOVSubscription>
      <Recipient>
        <Recipient>
          <Device>device-508</Device>
        </Recipient>
        <ProcessIdentifier>2353</ProcessIdentifier>
      </Recipient>
      <MonitoredPropertyReference>
        <ObjectIdentifier>device-510</ObjectIdentifier>
        <PropertyIdentifier>apduSegmentTimeout</PropertyIdentifier>
        <PropertyArrayIndex>1245</PropertyArrayIndex>
      </MonitoredPropertyReference>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <TimeRemaining>1246</TimeRemaining>
      <CovIncrement>1247.567749</CovIncrement>
    </BACnetCOVSubscription>
    <BACnetCOVSubscription>
      <Recipient>
        <Recipient>
          <Device>device-515</Device>
        </Recipient>
        <ProcessIdentifier>2360</ProcessIdentifier>
      </Recipient>
      <MonitoredPropertyReference>
        <ObjectIdentifier>device-517</ObjectIdentifier>
        <PropertyIdentifier>notificationClass</PropertyIdentifier>
        <PropertyArrayIndex>1252</PropertyArrayIndex>
      </MonitoredPropertyReference>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <TimeRemaining>1253</TimeRemaining>
      <CovIncrement>1254.567749</CovIncrement>
    </BACnetCOVSubscription>
  </SequenceOfBACnetCOVSubscription>
</Value>
```

SequenceOfBACnetCalendarEntry

This XML representation of complex data is used by the following objects/properties:

- **calendar.dateList**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetCalendarEntry">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetCalendarEntry"
type="BACnetCalendarEntry" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetCalendarEntry" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetCalendarEntry />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetCalendarEntry>
    <BACnetCalendarEntry>
      <Date>2017-12-20</Date>
    </BACnetCalendarEntry>
    <BACnetCalendarEntry>
      <Date>2018-12-20</Date>
    </BACnetCalendarEntry>
    <BACnetCalendarEntry>
      <Date>2019-12-20</Date>
    </BACnetCalendarEntry>
  </SequenceOfBACnetCalendarEntry>
</Value>
```

SequenceOfBACnetCredentialAuthenticationFactor

This XML representation of complex data is used by the following objects/properties:

- **accessCredential.authenticationFactors**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetCredentialAuthenticationFactor">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetCredentialAuthenticationFactor"
type="BACnetCredentialAuthenticationFactor" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetCredentialAuthenticationFactor" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetCredentialAuthenticationFactor />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetCredentialAuthenticationFactor>
    <BACnetCredentialAuthenticationFactor>
      <Disable>none</Disable>
      <AuthenticationFactor>
        <FormatType>error</FormatType>
        <FormatClass>1236</FormatClass>
        <Value>00010203040506070809</Value>
      </AuthenticationFactor>
    </BACnetCredentialAuthenticationFactor>
    <BACnetCredentialAuthenticationFactor>
      <Disable>disabledStolen</Disable>
      <AuthenticationFactor>
        <FormatType>simpleNumber32</FormatType>
        <FormatClass>1239</FormatClass>
        <Value>FF</Value>
      </AuthenticationFactor>
    </BACnetCredentialAuthenticationFactor>
    <BACnetCredentialAuthenticationFactor>
      <Disable>none</Disable>
      <AuthenticationFactor>
        <FormatType>abaTrack2</FormatType>
        <FormatClass>1242</FormatClass>
        <Value>FF</Value>
      </AuthenticationFactor>
    </BACnetCredentialAuthenticationFactor>
  </SequenceOfBACnetCredentialAuthenticationFactor>
</Value>
```

SequenceOfBACnetDestination

This XML representation of complex data is used by the following objects/properties:

- **notificationClass.recipientList**
- **notificationForwarder.recipientList**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetDestination">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetDestination"
        type="BACnetDestination" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetDestination" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetDestination />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetDestination>
    <BACnetDestination>
      <ValidDays>
        <monday>true</monday>
        <wednesday>true</wednesday>
        <friday>true</friday>
        <sunday>true</sunday>
      </ValidDays>
      <FromTime>22:39:15.980</FromTime>
      <ToTime>22:39:15.980</ToTime>
      <Recipient>
        <Device>device-503</Device>
      </Recipient>
      <ProcessIdentifier>2348</ProcessIdentifier>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <Transitions>
        <toOffnormal>true</toOffnormal>
        <toNormal>true</toNormal>
      </Transitions>
    </BACnetDestination>
    <BACnetDestination>
      <ValidDays>
        <monday>true</monday>
        <wednesday>true</wednesday>
        <friday>true</friday>
        <sunday>true</sunday>
      </ValidDays>
      <FromTime>22:39:15.980</FromTime>
      <ToTime>22:39:15.980</ToTime>
      <Recipient>
        <Device>device-507</Device>
      </Recipient>
      <ProcessIdentifier>2352</ProcessIdentifier>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <Transitions>
        <toOffnormal>true</toOffnormal>
        <toNormal>true</toNormal>
      </Transitions>
    </BACnetDestination>
    <BACnetDestination>
      <ValidDays>
        <monday>true</monday>
        <wednesday>true</wednesday>
        <friday>true</friday>
        <sunday>true</sunday>
      </ValidDays>
      <FromTime>22:39:15.980</FromTime>
      <ToTime>22:39:15.980</ToTime>
      <Recipient>
        <Device>device-511</Device>
      </Recipient>
      <ProcessIdentifier>2356</ProcessIdentifier>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <Transitions>
        <toOffnormal>true</toOffnormal>
        <toNormal>true</toNormal>
      </Transitions>
    </BACnetDestination>
  </SequenceOfBACnetDestination>
</Value>
```

SequenceOfBACnetDeviceObjectPropertyReference

This XML representation of complex data is used by the following objects/properties:

- **channel.listOfObjectPropertyReferences**
- **globalGroup.groupMembers**

- **schedule.listOfObjectPropertyReferences**
- **timer.listOfObjectPropertyReferences**
- **trendLogMultiple.logDeviceObjectProperty**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetDeviceObjectPropertyReference">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetDeviceObjectPropertyReference"
type="BACnetDeviceObjectPropertyReference" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetDeviceObjectPropertyReference" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetDeviceObjectPropertyReference />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetDeviceObjectPropertyReference>
    <BACnetDeviceObjectPropertyReference>
      <ObjectIdentifier>device-501</ObjectIdentifier>
      <PropertyIdentifier>ackRequired</PropertyIdentifier>
      <PropertyArrayIndex>1236</PropertyArrayIndex>
      <DeviceIdentifier>device-504</DeviceIdentifier>
    </BACnetDeviceObjectPropertyReference>
    <BACnetDeviceObjectPropertyReference>
      <ObjectIdentifier>device-505</ObjectIdentifier>
      <PropertyIdentifier>activeVtSessions</PropertyIdentifier>
      <PropertyArrayIndex>1240</PropertyArrayIndex>
      <DeviceIdentifier>device-508</DeviceIdentifier>
    </BACnetDeviceObjectPropertyReference>
    <BACnetDeviceObjectPropertyReference>
      <ObjectIdentifier>device-509</ObjectIdentifier>
      <PropertyIdentifier>allWritesSuccessful</PropertyIdentifier>
      <PropertyArrayIndex>1244</PropertyArrayIndex>
      <DeviceIdentifier>device-512</DeviceIdentifier>
    </BACnetDeviceObjectPropertyReference>
  </SequenceOfBACnetDeviceObjectPropertyReference>
</Value>
```

SequenceOfBACnetDeviceObjectReference

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.doorMembers**
- **accessPoint.accessDoors**
- **accessUser.credentials**
- **accessUser.memberOf**

- **accessUser.members**
- **accessZone.credentialsInZone**
- **accessZone.entryPoints**
- **accessZone.exitPoints**
- **lifeSafetyPoint.memberOf**
- **lifeSafetyZone.memberOf**
- **lifeSafetyZone.zoneMembers**
- **structuredView.subordinateList**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetDeviceObjectReference">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetDeviceObjectReference"
type="BACnetDeviceObjectReference" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetDeviceObjectReference" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetDeviceObjectReference />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetDeviceObjectReference>
    <BACnetDeviceObjectReference>
      <DeviceIdentifier>device-501</DeviceIdentifier>
      <ObjectIdentifier>device-502</ObjectIdentifier>
    </BACnetDeviceObjectReference>
    <BACnetDeviceObjectReference>
      <DeviceIdentifier>device-503</DeviceIdentifier>
      <ObjectIdentifier>device-504</ObjectIdentifier>
    </BACnetDeviceObjectReference>
    <BACnetDeviceObjectReference>
      <DeviceIdentifier>device-505</DeviceIdentifier>
      <ObjectIdentifier>device-506</ObjectIdentifier>
    </BACnetDeviceObjectReference>
  </SequenceOfBACnetDeviceObjectReference>
</Value>
```

SequenceOfBACnetDoorAlarmState

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.alarmValues**
- **accessDoor.faultValues**

- **accessDoor.maskedAlarmValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetDoorAlarmState">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetDoorAlarmState"
type="BACnetDoorAlarmState" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetDoorAlarmState" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetDoorAlarmState />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetDoorAlarmState>
    <BACnetDoorAlarmState>normal</BACnetDoorAlarmState>
    <BACnetDoorAlarmState>alarm</BACnetDoorAlarmState>
    <BACnetDoorAlarmState>doorOpenTooLong</BACnetDoorAlarmState>
  </SequenceOfBACnetDoorAlarmState>
</Value>
```

SequenceOfBACnetDoorStatus

This XML representation of complex data is used by the following objects/properties:

- **lift.carDoorStatus**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetDoorStatus">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetDoorStatus"
type="BACnetDoorStatus" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetDoorStatus" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetDoorStatus />
</Value>
```

Example 2:


```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetDoorStatus>
    <BACnetDoorStatus>closed</BACnetDoorStatus>
    <BACnetDoorStatus>opened</BACnetDoorStatus>
    <BACnetDoorStatus>unknown</BACnetDoorStatus>
  </SequenceOfBACnetDoorStatus>
</Value>
```

SequenceOfBACnetEscalatorFault

This XML representation of complex data is used by the following objects/properties:

- **escalator.faultSignals**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetEscalatorFault">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetEscalatorFault"
type="BACnetEscalatorFault" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetEscalatorFault" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetEscalatorFault />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetEscalatorFault>
    <BACnetEscalatorFault>controllerFault</BACnetEscalatorFault>
    <BACnetEscalatorFault>driveAndMotorFault</BACnetEscalatorFault>
    <BACnetEscalatorFault>mechanicalComponentFault</BACnetEscalatorFault>
  </SequenceOfBACnetEscalatorFault>
</Value>
```

SequenceOfBACnetEventLogRecord

This XML representation of complex data is used by the following objects/properties:

- **eventLog.logBuffer**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetEventLogRecord">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetEventLogRecord"
type="BACnetEventLogRecord" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetEventLogRecord" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetEventLogRecord />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetEventLogRecord>
    <BACnetEventLogRecord>
      <Timestamp>
        <Date>2017-12-20</Date>
        <Time>22:39:15.980</Time>
      </Timestamp>
      <LogDatum>
        <LogStatus>
          <logDisabled>true</logDisabled>
          <logInterrupted>true</logInterrupted>
        </LogStatus>
      </LogDatum>
    </BACnetEventLogRecord>
    <BACnetEventLogRecord>
      <Timestamp>
        <Date>2018-12-20</Date>
        <Time>22:39:15.980</Time>
      </Timestamp>
      <LogDatum>
        <LogStatus>
          <logDisabled>true</logDisabled>
          <logInterrupted>true</logInterrupted>
        </LogStatus>
      </LogDatum>
    </BACnetEventLogRecord>
    <BACnetEventLogRecord>
      <Timestamp>
        <Date>2019-12-20</Date>
        <Time>22:39:15.980</Time>
      </Timestamp>
      <LogDatum>
        <LogStatus>
          <logDisabled>true</logDisabled>
          <logInterrupted>true</logInterrupted>
        </LogStatus>
      </LogDatum>
    </BACnetEventLogRecord>
  </SequenceOfBACnetEventLogRecord>
</Value>
```

SequenceOfBACnetEventNotificationSubscription

This XML representation of complex data is used by the following objects/properties:

- **notificationForwarder.subscribedRecipients**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetEventNotificationSubscription">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetEventNotificationSubscription"
type="BACnetEventNotificationSubscription" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetEventNotificationSubscription" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetEventNotificationSubscription />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetEventNotificationSubscription>
    <BACnetEventNotificationSubscription>
      <Recipient>
        <Device>device-501</Device>
      </Recipient>
      <ProcessIdentifier>2346</ProcessIdentifier>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <TimeRemaining>1236</TimeRemaining>
    </BACnetEventNotificationSubscription>
    <BACnetEventNotificationSubscription>
      <Recipient>
        <Device>device-504</Device>
      </Recipient>
      <ProcessIdentifier>2349</ProcessIdentifier>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <TimeRemaining>1239</TimeRemaining>
    </BACnetEventNotificationSubscription>
    <BACnetEventNotificationSubscription>
      <Recipient>
        <Device>device-507</Device>
      </Recipient>
      <ProcessIdentifier>2352</ProcessIdentifier>
      <IssueConfirmedNotifications>true</IssueConfirmedNotifications>
      <TimeRemaining>1242</TimeRemaining>
    </BACnetEventNotificationSubscription>
  </SequenceOfBACnetEventNotificationSubscription>
</Value>
```

SequenceOfBACnetEventParameter_Extended_Unnamed

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="SequenceOfBACnetEventParameter_Extended_Unnamed">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Unnamed"
type="BACnetEventParameter_Extended_Unnamed" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<SequenceOfBACnetEventParameter_Extended_Unnamed />
```

Example 2:

```
<SequenceOfBACnetEventParameter_Extended_Unnamed>
  <Unnamed>
    <Null />
  </Unnamed>
  <Unnamed>
    <Null />
  </Unnamed>
  <Unnamed>
    <Null />
  </Unnamed>
  <Unnamed>
    <Null />
  </Unnamed>
</SequenceOfBACnetEventParameter_Extended_Unnamed>
```

SequenceOfBACnetFDTEEntry

This XML representation of complex data is used by the following objects/properties:

- **networkPort.bbmdForeignDeviceTable**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetFDTEEntry">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetFDTEEntry" type="BACnetFDTEEntry" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetFDTEEntry" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetFDTEEntry />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetFDTEEntry>
    <BACnetFDTEEntry>
      <BacnetIpAddress>00010203040506070809</BacnetIpAddress>
      <TimeToLive>7634</TimeToLive>
      <RemainingTimeToLive>7635</RemainingTimeToLive>
    </BACnetFDTEEntry>
    <BACnetFDTEEntry>
      <BacnetIpAddress>FF</BacnetIpAddress>
      <TimeToLive>7636</TimeToLive>
      <RemainingTimeToLive>7637</RemainingTimeToLive>
    </BACnetFDTEEntry>
    <BACnetFDTEEntry>
      <BacnetIpAddress>FF</BacnetIpAddress>
      <TimeToLive>7638</TimeToLive>
      <RemainingTimeToLive>7639</RemainingTimeToLive>
    </BACnetFDTEEntry>
  </SequenceOfBACnetFDTEEntry>
</Value>
```

SequenceOfBACnetFaultParameter_FaultExtended_Unnamed

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="SequenceOfBACnetFaultParameter_FaultExtended_Unnamed">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Unnamed"
type="BACnetFaultParameter_FaultExtended_Unnamed" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<SequenceOfBACnetFaultParameter_FaultExtended_Unnamed />
```

Example 2:

```
<SequenceOfBACnetFaultParameter_FaultExtended_Unnamed>
  <Unnamed>
    <Null />
  </Unnamed>
  <Unnamed>
    <Null />
  </Unnamed>
  <Unnamed>
    <Null />
  </Unnamed>
</SequenceOfBACnetFaultParameter_FaultExtended_Unnamed>
```

SequenceOfBACnetKeyIdentifier

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="SequenceOfBACnetKeyIdentifier">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetKeyIdentifier"
type="BACnetKeyIdentifier" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<SequenceOfBACnetKeyIdentifier />
```

Example 2:

```
<SequenceOfBACnetKeyIdentifier>
  <BACnetKeyIdentifier>
    <Algorithm>33</Algorithm>
    <KeyId>34</KeyId>
  </BACnetKeyIdentifier>
  <BACnetKeyIdentifier>
    <Algorithm>35</Algorithm>
    <KeyId>36</KeyId>
  </BACnetKeyIdentifier>
  <BACnetKeyIdentifier>
    <Algorithm>37</Algorithm>
    <KeyId>38</KeyId>
  </BACnetKeyIdentifier>
</SequenceOfBACnetKeyIdentifier>
```

SequenceOfBACnetLandingCallStatus

This XML representation of complex data is used by the following objects/properties:

- **elevatorGroup.landingCalls**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetLandingCallStatus">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetLandingCallStatus"
type="BACnetLandingCallStatus" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetLandingCallStatus" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLandingCallStatus />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLandingCallStatus>
    <BACnetLandingCallStatus>
      <FloorNumber>33</FloorNumber>
      <Command>
        <Direction>none</Direction>
      </Command>
      <FloorText>Abcde</FloorText>
    </BACnetLandingCallStatus>
    <BACnetLandingCallStatus>
      <FloorNumber>35</FloorNumber>
      <Command>
        <Direction>up</Direction>
      </Command>
      <FloorText>Abcde</FloorText>
    </BACnetLandingCallStatus>
    <BACnetLandingCallStatus>
      <FloorNumber>37</FloorNumber>
      <Command>
        <Direction>upAndDown</Direction>
      </Command>
      <FloorText>Abcde</FloorText>
    </BACnetLandingCallStatus>
  </SequenceOfBACnetLandingCallStatus>
</Value>
```

SequenceOfBACnetLandingDoorStatus

This XML representation of complex data is used by the following objects/properties:

- **lift.landingDoorStatus**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetLandingDoorStatus">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetLandingDoorStatus"
type="BACnetLandingDoorStatus" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetLandingDoorStatus" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLandingDoorStatus />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLandingDoorStatus>
    <BACnetLandingDoorStatus>
      <LandingDoors />
    </BACnetLandingDoorStatus>
    <BACnetLandingDoorStatus>
      <LandingDoors>
        <LandingDoors_Item>
          <FloorNumber>33</FloorNumber>
          <DoorStatus>opened</DoorStatus>
        </LandingDoors_Item>
      </LandingDoors>
    </BACnetLandingDoorStatus>
    <BACnetLandingDoorStatus>
      <LandingDoors>
        <LandingDoors_Item>
          <FloorNumber>35</FloorNumber>
          <DoorStatus>doorFault</DoorStatus>
        </LandingDoors_Item>
      </LandingDoors>
    </BACnetLandingDoorStatus>
  </SequenceOfBACnetLandingDoorStatus>
</Value>
```

Example 3:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLandingDoorStatus>
    <BACnetLandingDoorStatus>
      <LandingDoors>
        <LandingDoors_Item>
          <FloorNumber>37</FloorNumber>
          <DoorStatus>none</DoorStatus>
        </LandingDoors_Item>
        <LandingDoors_Item>
          <FloorNumber>39</FloorNumber>
          <DoorStatus>opening</DoorStatus>
        </LandingDoors_Item>
        <LandingDoors_Item>
          <FloorNumber>41</FloorNumber>
          <DoorStatus>limitedOpened</DoorStatus>
        </LandingDoors_Item>
      </LandingDoors>
    </BACnetLandingDoorStatus>
    <BACnetLandingDoorStatus>
      <LandingDoors>
        <LandingDoors_Item>
          <FloorNumber>43</FloorNumber>
          <DoorStatus>opened</DoorStatus>
        </LandingDoors_Item>
      </LandingDoors>
    </BACnetLandingDoorStatus>
    <BACnetLandingDoorStatus>
      <LandingDoors>
        <LandingDoors_Item>
          <FloorNumber>45</FloorNumber>
          <DoorStatus>doorFault</DoorStatus>
        </LandingDoors_Item>
      </LandingDoors>
    </BACnetLandingDoorStatus>
  </SequenceOfBACnetLandingDoorStatus>
</Value>
```

SequenceOfBACnetLifeSafetyMode

This XML representation of complex data is used by the following objects/properties:

- **lifeSafetyPoint.acceptedModes**

- **lifeSafetyZone.acceptedModes**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetLifeSafetyMode">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetLifeSafetyMode"
type="BACnetLifeSafetyMode" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetLifeSafetyMode" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLifeSafetyMode />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLifeSafetyMode>
    <BACnetLifeSafetyMode>off</BACnetLifeSafetyMode>
    <BACnetLifeSafetyMode>on</BACnetLifeSafetyMode>
    <BACnetLifeSafetyMode>test</BACnetLifeSafetyMode>
  </SequenceOfBACnetLifeSafetyMode>
</Value>
```

SequenceOfBACnetLifeSafetyState

This XML representation of complex data is used by the following objects/properties:

- **lifeSafetyPoint.alarmValues**
- **lifeSafetyPoint.faultValues**
- **lifeSafetyPoint.lifeSafetyAlarmValues**
- **lifeSafetyZone.alarmValues**
- **lifeSafetyZone.faultValues**
- **lifeSafetyZone.lifeSafetyAlarmValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetLifeSafetyState">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetLifeSafetyState"
type="BACnetLifeSafetyState" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetLifeSafetyState" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLifeSafetyState />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLifeSafetyState>
    <BACnetLifeSafetyState>quiet</BACnetLifeSafetyState>
    <BACnetLifeSafetyState>preAlarm</BACnetLifeSafetyState>
    <BACnetLifeSafetyState>alarm</BACnetLifeSafetyState>
  </SequenceOfBACnetLifeSafetyState>
</Value>
```

SequenceOfBACnetLiftCarCallList

This XML representation of complex data is used by the following objects/properties:

- **lift.registeredCarCall**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetLiftCarCallList">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetLiftCarCallList"
type="BACnetLiftCarCallList" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetLiftCarCallList" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLiftCarCallList />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLiftCarCallList>
    <BACnetLiftCarCallList>
      <FloorNumbers />
    </BACnetLiftCarCallList>
    <BACnetLiftCarCallList>
      <FloorNumbers>
        <Unsigned8>33</Unsigned8>
      </FloorNumbers>
    </BACnetLiftCarCallList>
    <BACnetLiftCarCallList>
      <FloorNumbers>
        <Unsigned8>34</Unsigned8>
      </FloorNumbers>
    </BACnetLiftCarCallList>
  </SequenceOfBACnetLiftCarCallList>
</Value>
```

Example 3:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLiftCarCallList>
    <BACnetLiftCarCallList>
      <FloorNumbers>
        <Unsigned8>35</Unsigned8>
        <Unsigned8>36</Unsigned8>
        <Unsigned8>37</Unsigned8>
      </FloorNumbers>
    </BACnetLiftCarCallList>
    <BACnetLiftCarCallList>
      <FloorNumbers>
        <Unsigned8>38</Unsigned8>
      </FloorNumbers>
    </BACnetLiftCarCallList>
    <BACnetLiftCarCallList>
      <FloorNumbers>
        <Unsigned8>39</Unsigned8>
      </FloorNumbers>
    </BACnetLiftCarCallList>
  </SequenceOfBACnetLiftCarCallList>
</Value>
```

SequenceOfBACnetLiftCarDoorCommand

This XML representation of complex data is used by the following objects/properties:

- **lift.carDoorCommand**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetLiftCarDoorCommand">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetLiftCarDoorCommand"
type="BACnetLiftCarDoorCommand" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetLiftCarDoorCommand" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLiftCarDoorCommand />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLiftCarDoorCommand>
    <BACnetLiftCarDoorCommand>none</BACnetLiftCarDoorCommand>
    <BACnetLiftCarDoorCommand>open</BACnetLiftCarDoorCommand>
    <BACnetLiftCarDoorCommand>close</BACnetLiftCarDoorCommand>
  </SequenceOfBACnetLiftCarDoorCommand>
</Value>
```

SequenceOfBACnetLiftFault

This XML representation of complex data is used by the following objects/properties:

- **lift.faultSignals**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetLiftFault">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetLiftFault"
type="BACnetLiftFault" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetLiftFault" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLiftFault />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLiftFault>
    <BACnetLiftFault>controllerFault</BACnetLiftFault>
    <BACnetLiftFault>driveAndMotorFault</BACnetLiftFault>
    <BACnetLiftFault>governorAndSafetyGearFault</BACnetLiftFault>
  </SequenceOfBACnetLiftFault>
</Value>
```

SequenceOfBACnetLogData_Unnamed

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="SequenceOfBACnetLogData_Unnamed">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Unnamed" type="BACnetLogData_Unnamed" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<SequenceOfBACnetLogData_Unnamed />
```

Example 2:

```
<SequenceOfBACnetLogData_Unnamed>
  <Unnamed>
    <BooleanValue>true</BooleanValue>
  </Unnamed>
  <Unnamed>
    <BooleanValue>true</BooleanValue>
  </Unnamed>
  <Unnamed>
    <BooleanValue>true</BooleanValue>
  </Unnamed>
</SequenceOfBACnetLogData_Unnamed>
```

SequenceOfBACnetLogMultipleRecord

This XML representation of complex data is used by the following objects/properties:

- **trendLogMultiple.logBuffer**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetLogMultipleRecord">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetLogMultipleRecord"
type="BACnetLogMultipleRecord" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetLogMultipleRecord" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLogMultipleRecord />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLogMultipleRecord>
    <BACnetLogMultipleRecord>
      <Timestamp>
        <Date>2017-12-20</Date>
        <Time>22:39:15.980</Time>
      </Timestamp>
      <LogData>
        <LogStatus>
          <logDisabled>true</logDisabled>
          <logInterrupted>true</logInterrupted>
        </LogStatus>
      </LogData>
    </BACnetLogMultipleRecord>
    <BACnetLogMultipleRecord>
      <Timestamp>
        <Date>2018-12-20</Date>
        <Time>22:39:15.980</Time>
      </Timestamp>
      <LogData>
        <LogStatus>
          <logDisabled>true</logDisabled>
          <logInterrupted>true</logInterrupted>
        </LogStatus>
      </LogData>
    </BACnetLogMultipleRecord>
    <BACnetLogMultipleRecord>
      <Timestamp>
        <Date>2019-12-20</Date>
        <Time>22:39:15.980</Time>
      </Timestamp>
      <LogData>
        <LogStatus>
          <logDisabled>true</logDisabled>
          <logInterrupted>true</logInterrupted>
        </LogStatus>
      </LogData>
    </BACnetLogMultipleRecord>
  </SequenceOfBACnetLogMultipleRecord>
</Value>
```

SequenceOfBACnetLogRecord

This XML representation of complex data is used by the following objects/properties:

- **trendLog.logBuffer**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetLogRecord">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetLogRecord"
type="BACnetLogRecord" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetLogRecord" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLogRecord />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetLogRecord>
    <BACnetLogRecord>
      <Timestamp>
        <Date>2017-12-20</Date>
        <Time>22:39:15.980</Time>
      </Timestamp>
      <LogDatum>
        <LogStatus>
          <logDisabled>true</logDisabled>
          <logInterrupted>true</logInterrupted>
        </LogStatus>
      </LogDatum>
      <StatusFlags>
        <inAlarm>true</inAlarm>
        <overridden>true</overridden>
      </StatusFlags>
    </BACnetLogRecord>
    <BACnetLogRecord>
      <Timestamp>
        <Date>2018-12-20</Date>
        <Time>22:39:15.980</Time>
      </Timestamp>
      <LogDatum>
        <LogStatus>
          <logDisabled>true</logDisabled>
          <logInterrupted>true</logInterrupted>
        </LogStatus>
      </LogDatum>
      <StatusFlags>
        <inAlarm>true</inAlarm>
        <overridden>true</overridden>
      </StatusFlags>
    </BACnetLogRecord>
    <BACnetLogRecord>
      <Timestamp>
        <Date>2019-12-20</Date>
        <Time>22:39:15.980</Time>
      </Timestamp>
      <LogDatum>
        <LogStatus>
          <logDisabled>true</logDisabled>
          <logInterrupted>true</logInterrupted>
        </LogStatus>
      </LogDatum>
      <StatusFlags>
        <inAlarm>true</inAlarm>
        <overridden>true</overridden>
      </StatusFlags>
    </BACnetLogRecord>
  </SequenceOfBACnetLogRecord>
</Value>
```

SequenceOfBACnetNameValue

This XML representation of complex data is used by the following objects/properties:

- **accessCredential.tags**
- **accessDoor.tags**
- **accessPoint.tags**

- **accessRights.tags**
- **accessUser.tags**
- **accessZone.tags**
- **accumulator.tags**
- **alertEnrollment.tags**
- **analogInput.tags**
- **analogOutput.tags**
- **analogValue.tags**
- **averaging.tags**
- **binaryInput.tags**
- **binaryLightingOutput.tags**
- **binaryOutput.tags**
- **binaryValue.tags**
- **bitstringValue.tags**
- **calendar.tags**
- **channel.tags**
- **characterstringValue.tags**
- **command.tags**
- **credentialDataInput.tags**
- **dateValue.tags**
- **datepatternValue.tags**
- **datetimeValue.tags**
- **datetimepatternValue.tags**
- **device.tags**
- **elevatorGroup.tags**
- **escalator.tags**
- **eventEnrollment.tags**
- **eventLog.tags**
- **file.tags**
- **globalGroup.tags**
- **group.tags**
- **integerValue.tags**

- **largeAnalogValue.tags**
- **lifeSafetyPoint.tags**
- **lifeSafetyZone.tags**
- **lift.tags**
- **lightingOutput.tags**
- **loadControl.tags**
- **loop.tags**
- **multiStateInput.tags**
- **multiStateOutput.tags**
- **multiStateValue.tags**
- **networkPort.tags**
- **networkSecurity.tags**
- **notificationClass.tags**
- **notificationForwarder.tags**
- **octetstringValue.tags**
- **positiveIntegerValue.tags**
- **program.tags**
- **pulseConverter.tags**
- **schedule.tags**
- **structuredView.tags**
- **timeValue.tags**
- **timepatternValue.tags**
- **timer.tags**
- **trendLog.tags**
- **trendLogMultiple.tags**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetNameValue">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetNameValue"
type="BACnetNameValue" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetNameValue" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetNameValue />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetNameValue>
    <BACnetNameValue>
      <Name>Abcde</Name>
      <Value>
        <Boolean>true</Boolean>
      </Value>
    </BACnetNameValue>
    <BACnetNameValue>
      <Name>Abcde</Name>
      <Value>
        <Boolean>true</Boolean>
      </Value>
    </BACnetNameValue>
    <BACnetNameValue>
      <Name>Abcde</Name>
      <Value>
        <Boolean>true</Boolean>
      </Value>
    </BACnetNameValue>
  </SequenceOfBACnetNameValue>
</Value>
```

SequenceOfBACnetNameValueCollection

This XML representation of complex data is used by the following objects/properties:

- **structuredView.subordinateTags**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetNameValueCollection">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetNameValueCollection"
type="BACnetNameValueCollection" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetNameValueCollection" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetNameValueCollection />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetNameValueCollection>
    <BACnetNameValueCollection>
      <Members />
    </BACnetNameValueCollection>
    <BACnetNameValueCollection>
      <Members>
        <BACnetNameValue>
          <Name>Abcde</Name>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetNameValue>
      </Members>
    </BACnetNameValueCollection>
    <BACnetNameValueCollection>
      <Members>
        <BACnetNameValue>
          <Name>Abcde</Name>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetNameValue>
      </Members>
    </BACnetNameValueCollection>
  </SequenceOfBACnetNameValueCollection>
</Value>
```

Example 3:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetNameValueCollection>
    <BACnetNameValueCollection>
      <Members>
        <BACnetNameValue>
          <Name>Abcde</Name>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetNameValue>
        <BACnetNameValue>
          <Name>Abcde</Name>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetNameValue>
        <BACnetNameValue>
          <Name>Abcde</Name>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetNameValue>
      </Members>
    </BACnetNameValueCollection>
    <BACnetNameValueCollection>
      <Members>
        <BACnetNameValue>
          <Name>Abcde</Name>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetNameValue>
      </Members>
    </BACnetNameValueCollection>
    <BACnetNameValueCollection>
      <Members>
        <BACnetNameValue>
          <Name>Abcde</Name>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetNameValue>
      </Members>
    </BACnetNameValueCollection>
  </SequenceOfBACnetNameValueCollection>
</Value>

```

SequenceOfBACnetNetworkSecurityPolicy

This XML representation of complex data is used by the following objects/properties:

- **networkSecurity.networkAccessSecurityPolicies**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetNetworkSecurityPolicy">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetNetworkSecurityPolicy"
type="BACnetNetworkSecurityPolicy" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetNetworkSecurityPolicy" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetNetworkSecurityPolicy />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetNetworkSecurityPolicy>
    <BACnetNetworkSecurityPolicy>
      <PortId>33</PortId>
      <SecurityLevel>plainTrusted</SecurityLevel>
    </BACnetNetworkSecurityPolicy>
    <BACnetNetworkSecurityPolicy>
      <PortId>35</PortId>
      <SecurityLevel>encryptedTrusted</SecurityLevel>
    </BACnetNetworkSecurityPolicy>
    <BACnetNetworkSecurityPolicy>
      <PortId>37</PortId>
      <SecurityLevel>plainTrusted</SecurityLevel>
    </BACnetNetworkSecurityPolicy>
  </SequenceOfBACnetNetworkSecurityPolicy>
</Value>
```

SequenceOfBACnetNodeType

This XML representation of complex data is used by the following objects/properties:

- **structuredView.subordinateNodeTypes**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetNodeType">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetNodeType" type="BACnetNodeType" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetNodeType" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetNodeType />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetNodeType>
    <BACnetNodeType>unknown</BACnetNodeType>
    <BACnetNodeType>system</BACnetNodeType>
    <BACnetNodeType>network</BACnetNodeType>
  </SequenceOfBACnetNodeType>
</Value>
```

SequenceOfBACnetNotificationParameters_Extended_Unnamed

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="SequenceOfBACnetNotificationParameters_Extended_Unnamed">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Unnamed"
type="BACnetNotificationParameters_Extended_Unnamed" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<SequenceOfBACnetNotificationParameters_Extended_Unnamed />
```

Example 2:

```
<SequenceOfBACnetNotificationParameters_Extended_Unnamed>
  <Unnamed>
    <Null />
  </Unnamed>
  <Unnamed>
    <Null />
  </Unnamed>
  <Unnamed>
    <Null />
  </Unnamed>
</SequenceOfBACnetNotificationParameters_Extended_Unnamed>
```

SequenceOfBACnetObjectIdentifier

This XML representation of complex data is used by the following objects/properties:

- **device.configurationFiles**
- **device.objectList**
- **device.structuredObjectList**
- **elevatorGroup.groupMembers**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetObjectIdentifier">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetObjectIdentifier"
type="BACnetObjectIdentifier" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetObjectIdentifier" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetObjectIdentifier />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetObjectIdentifier>
    <BACnetObjectIdentifier>device-501</BACnetObjectIdentifier>
    <BACnetObjectIdentifier>device-502</BACnetObjectIdentifier>
    <BACnetObjectIdentifier>device-503</BACnetObjectIdentifier>
  </SequenceOfBACnetObjectIdentifier>
</Value>
```

SequenceOfBACnetOptionalCharacterString

This XML representation of complex data is used by the following objects/properties:

- **characterstringValue.alarmValues**
- **characterstringValue.faultValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetOptionalCharacterString">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetOptionalCharacterString"
type="BACnetOptionalCharacterString" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetOptionalCharacterString" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetOptionalCharacterString />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetOptionalCharacterString>
    <BACnetOptionalCharacterString>
      <Null />
    </BACnetOptionalCharacterString>
    <BACnetOptionalCharacterString>
      <Null />
    </BACnetOptionalCharacterString>
    <BACnetOptionalCharacterString>
      <Null />
    </BACnetOptionalCharacterString>
  </SequenceOfBACnetOptionalCharacterString>
</Value>
```

SequenceOfBACnetPortPermission

This XML representation of complex data is used by the following objects/properties:

- **notificationForwarder.portFilter**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetPortPermission">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetPortPermission"
        type="BACnetPortPermission" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetPortPermission" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetPortPermission />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetPortPermission>
    <BACnetPortPermission>
      <PortId>33</PortId>
      <Enabled>true</Enabled>
    </BACnetPortPermission>
    <BACnetPortPermission>
      <PortId>34</PortId>
      <Enabled>true</Enabled>
    </BACnetPortPermission>
    <BACnetPortPermission>
      <PortId>35</PortId>
      <Enabled>true</Enabled>
    </BACnetPortPermission>
  </SequenceOfBACnetPortPermission>
</Value>
```

SequenceOfBACnetPropertyAccessResult

This XML representation of complex data is used by the following objects/properties:

- `globalGroup.presentValue`

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetPropertyAccessResult">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetPropertyAccessResult"
type="BACnetPropertyAccessResult" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetPropertyAccessResult" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetPropertyAccessResult />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetPropertyAccessResult>
    <BACnetPropertyAccessResult>
      <ObjectIdentifier>device-501</ObjectIdentifier>
      <PropertyIdentifier>ackRequired</PropertyIdentifier>
      <PropertyArrayIndex>1236</PropertyArrayIndex>
      <DeviceIdentifier>device-504</DeviceIdentifier>
      <AccessResult>
        <PropertyValue>
          <Boolean>true</Boolean>
        </PropertyValue>
      </AccessResult>
    </BACnetPropertyAccessResult>
    <BACnetPropertyAccessResult>
      <ObjectIdentifier>device-505</ObjectIdentifier>
      <PropertyIdentifier>activeVtSessions</PropertyIdentifier>
      <PropertyArrayIndex>1240</PropertyArrayIndex>
      <DeviceIdentifier>device-508</DeviceIdentifier>
      <AccessResult>
        <PropertyValue>
          <BACnetVTSession>
            <LocalVtSessionId>41</LocalVtSessionId>
            <RemoteVtSessionId>42</RemoteVtSessionId>
            <RemoteVtAddress>
              <NetworkNumber>7644</NetworkNumber>
              <MacAddress>FF</MacAddress>
            </RemoteVtAddress>
          </BACnetVTSession>
        </PropertyValue>
      </AccessResult>
    </BACnetPropertyAccessResult>
    <BACnetPropertyAccessResult>
      <ObjectIdentifier>device-512</ObjectIdentifier>
      <PropertyIdentifier>applicationSoftwareVersion</PropertyIdentifier>
      <PropertyArrayIndex>1247</PropertyArrayIndex>
      <DeviceIdentifier>device-515</DeviceIdentifier>
      <AccessResult>
        <PropertyValue>
          <CharacterString>Abcde</CharacterString>
        </PropertyValue>
      </AccessResult>
    </BACnetPropertyAccessResult>
  </SequenceOfBACnetPropertyAccessResult>
</Value>
```

SequenceOfBACnetPropertyIdentifier

This XML representation of complex data is used by the following objects/properties:

- **accessCredential.propertyList**
- **accessDoor.propertyList**
- **accessPoint.propertyList**
- **accessRights.propertyList**
- **accessUser.propertyList**
- **accessZone.propertyList**
- **accumulator.propertyList**
- **alertEnrollment.propertyList**
- **analogInput.propertyList**

- **analogOutput.propertyList**
- **analogValue.propertyList**
- **averaging.propertyList**
- **binaryInput.propertyList**
- **binaryLightingOutput.propertyList**
- **binaryOutput.propertyList**
- **binaryValue.propertyList**
- **bitstringValue.propertyList**
- **calendar.propertyList**
- **channel.propertyList**
- **characterstringValue.propertyList**
- **command.propertyList**
- **credentialDataInput.propertyList**
- **dateValue.propertyList**
- **datepatternValue.propertyList**
- **datetimeValue.propertyList**
- **datetimepatternValue.propertyList**
- **device.propertyList**
- **elevatorGroup.propertyList**
- **escalator.propertyList**
- **eventEnrollment.propertyList**
- **eventLog.propertyList**
- **file.propertyList**
- **globalGroup.propertyList**
- **group.propertyList**
- **integerValue.propertyList**
- **largeAnalogValue.propertyList**
- **lifeSafetyPoint.propertyList**
- **lifeSafetyZone.propertyList**
- **lift.propertyList**
- **lightingOutput.propertyList**
- **loadControl.propertyList**

- **loop.propertyList**
- **multiStateInput.propertyList**
- **multiStateOutput.propertyList**
- **multiStateValue.propertyList**
- **networkPort.propertyList**
- **networkSecurity.propertyList**
- **notificationClass.propertyList**
- **notificationForwarder.propertyList**
- **octetstringValue.propertyList**
- **positiveIntegerValue.propertyList**
- **program.propertyList**
- **pulseConverter.propertyList**
- **schedule.propertyList**
- **structuredView.propertyList**
- **timeValue.propertyList**
- **timepatternValue.propertyList**
- **timer.propertyList**
- **trendLog.propertyList**
- **trendLogMultiple.propertyList**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetPropertyIdentifier">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetPropertyIdentifier"
type="BACnetPropertyIdentifier" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetPropertyIdentifier" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetPropertyIdentifier />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetPropertyIdentifier>
    <BACnetPropertyIdentifier>ackedTransitions</BACnetPropertyIdentifier>
    <BACnetPropertyIdentifier>ackRequired</BACnetPropertyIdentifier>
    <BACnetPropertyIdentifier>action</BACnetPropertyIdentifier>
  </SequenceOfBACnetPropertyIdentifier>
</Value>
```

SequenceOfBACnetPropertyReference

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="SequenceOfBACnetPropertyReference">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetPropertyReference"
type="BACnetPropertyReference" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<SequenceOfBACnetPropertyReference />
```

Example 2:

```
<SequenceOfBACnetPropertyReference>
  <BACnetPropertyReference>
    <PropertyIdentifier>ackedTransitions</PropertyIdentifier>
    <PropertyArrayIndex>1235</PropertyArrayIndex>
  </BACnetPropertyReference>
  <BACnetPropertyReference>
    <PropertyIdentifier>action</PropertyIdentifier>
    <PropertyArrayIndex>1237</PropertyArrayIndex>
  </BACnetPropertyReference>
  <BACnetPropertyReference>
    <PropertyIdentifier>activeText</PropertyIdentifier>
    <PropertyArrayIndex>1239</PropertyArrayIndex>
  </BACnetPropertyReference>
</SequenceOfBACnetPropertyReference>
```

SequenceOfBACnetPropertyStates

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="SequenceOfBACnetPropertyStates">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetPropertyStates"
type="BACnetPropertyStates" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<SequenceOfBACnetPropertyStates />
```

Example 2:

```

<SequenceOfBACnetPropertyStates>
  <BACnetPropertyStates>
    <BooleanValue>true</BooleanValue>
  </BACnetPropertyStates>
  <BACnetPropertyStates>
    <BooleanValue>true</BooleanValue>
  </BACnetPropertyStates>
  <BACnetPropertyStates>
    <BooleanValue>true</BooleanValue>
  </BACnetPropertyStates>
</SequenceOfBACnetPropertyStates>

```

SequenceOfBACnetPropertyValue

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```

<xs:complexType name="SequenceOfBACnetPropertyValue">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetPropertyValue"
type="BACnetPropertyValue" />
  </xs:sequence>
</xs:complexType>

```

Examples

Example 1:

```
<SequenceOfBACnetPropertyValue />
```

Example 2:

```

<SequenceOfBACnetPropertyValue>
  <BACnetPropertyValue>
    <PropertyIdentifier>ackedTransitions</PropertyIdentifier>
    <PropertyArrayIndex>1235</PropertyArrayIndex>
    <PropertyValue>
      <Boolean>true</Boolean>
    </PropertyValue>
    <Priority>1236</Priority>
  </BACnetPropertyValue>
  <BACnetPropertyValue>
    <PropertyIdentifier>actionText</PropertyIdentifier>
    <PropertyArrayIndex>1238</PropertyArrayIndex>
    <PropertyValue>
      <Boolean>true</Boolean>
    </PropertyValue>
    <Priority>1239</Priority>
  </BACnetPropertyValue>
  <BACnetPropertyValue>
    <PropertyIdentifier>alarmValue</PropertyIdentifier>
    <PropertyArrayIndex>1241</PropertyArrayIndex>
    <PropertyValue>
      <Boolean>true</Boolean>
    </PropertyValue>
    <Priority>1242</Priority>
  </BACnetPropertyValue>
</SequenceOfBACnetPropertyValue>

```

SequenceOfBACnetRecipient

This XML representation of complex data is used by the following objects/properties:

- **device.restartNotificationRecipients**
- **device.timeSynchronizationRecipients**
- **device.utcTimeSynchronizationRecipients**

- **globalGroup.covuRecipients**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetRecipient">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetRecipient"
type="BACnetRecipient" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetRecipient" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetRecipient />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetRecipient>
    <BACnetRecipient>
      <Device>device-501</Device>
    </BACnetRecipient>
    <BACnetRecipient>
      <Device>device-502</Device>
    </BACnetRecipient>
    <BACnetRecipient>
      <Device>device-503</Device>
    </BACnetRecipient>
  </SequenceOfBACnetRecipient>
</Value>
```

SequenceOfBACnetRelationship

This XML representation of complex data is used by the following objects/properties:

- **structuredView.subordinateRelationships**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetRelationship">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetRelationship"
type="BACnetRelationship" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetRelationship" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetRelationship />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetRelationship>
    <BACnetRelationship>unknown</BACnetRelationship>
    <BACnetRelationship>default</BACnetRelationship>
    <BACnetRelationship>contains</BACnetRelationship>
  </SequenceOfBACnetRelationship>
</Value>
```

SequenceOfBACnetRouterEntry

This XML representation of complex data is used by the following objects/properties:

- **networkPort.routingTable**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetRouterEntry">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetRouterEntry"
type="BACnetRouterEntry" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetRouterEntry" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetRouterEntry />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetRouterEntry>
    <BACnetRouterEntry>
      <NetworkNumber>7634</NetworkNumber>
      <MacAddress>00010203040506070809</MacAddress>
      <Status>busy</Status>
      <PerformanceIndex>35</PerformanceIndex>
    </BACnetRouterEntry>
    <BACnetRouterEntry>
      <NetworkNumber>7637</NetworkNumber>
      <MacAddress>FF</MacAddress>
      <Status>busy</Status>
      <PerformanceIndex>38</PerformanceIndex>
    </BACnetRouterEntry>
    <BACnetRouterEntry>
      <NetworkNumber>7640</NetworkNumber>
      <MacAddress>FF</MacAddress>
      <Status>busy</Status>
      <PerformanceIndex>41</PerformanceIndex>
    </BACnetRouterEntry>
  </SequenceOfBACnetRouterEntry>
</Value>
```


SequenceOfBACnetSpecialEvent

This XML representation of complex data is used by the following objects/properties:

- **schedule.exceptionSchedule**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetSpecialEvent">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetSpecialEvent"
type="BACnetSpecialEvent" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetSpecialEvent" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetSpecialEvent />
</Value>
```

Example 2:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfBACnetSpecialEvent>
    <BACnetSpecialEvent>
      <Period>
        <CalendarEntry>
          <Date>2017-12-20</Date>
        </CalendarEntry>
      </Period>
      <ListOfTimeValues />
      <EventPriority>1235</EventPriority>
    </BACnetSpecialEvent>
    <BACnetSpecialEvent>
      <Period>
        <CalendarEntry>
          <Date>2019-12-20</Date>
        </CalendarEntry>
      </Period>
      <ListOfTimeValues>
        <BACnetTimeValue>
          <Time>22:39:15.980</Time>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetTimeValue>
      </ListOfTimeValues>
      <EventPriority>1238</EventPriority>
    </BACnetSpecialEvent>
    <BACnetSpecialEvent>
      <Period>
        <CalendarEntry>
          <Date>2022-12-20</Date>
        </CalendarEntry>
      </Period>
      <ListOfTimeValues>
        <BACnetTimeValue>
          <Time>22:39:15.980</Time>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetTimeValue>
      </ListOfTimeValues>
      <EventPriority>1241</EventPriority>
    </BACnetSpecialEvent>
  </SequenceOfBACnetSpecialEvent>
</Value>

```

Example 3:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetSpecialEvent>
    <BACnetSpecialEvent>
      <Period>
        <CalendarEntry>
          <Date>2025-12-20</Date>
        </CalendarEntry>
      </Period>
      <ListOfTimeValues>
        <BACnetTimeValue>
          <Time>22:39:15.980</Time>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetTimeValue>
        <BACnetTimeValue>
          <Time>22:39:15.980</Time>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetTimeValue>
        <BACnetTimeValue>
          <Time>22:39:15.980</Time>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetTimeValue>
      </ListOfTimeValues>
      <EventPriority>1246</EventPriority>
    </BACnetSpecialEvent>
    <BACnetSpecialEvent>
      <Period>
        <CalendarEntry>
          <Date>2030-12-20</Date>
        </CalendarEntry>
      </Period>
      <ListOfTimeValues>
        <BACnetTimeValue>
          <Time>22:39:15.980</Time>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetTimeValue>
      </ListOfTimeValues>
      <EventPriority>1249</EventPriority>
    </BACnetSpecialEvent>
    <BACnetSpecialEvent>
      <Period>
        <CalendarEntry>
          <Date>2033-12-20</Date>
        </CalendarEntry>
      </Period>
      <ListOfTimeValues>
        <BACnetTimeValue>
          <Time>22:39:15.980</Time>
          <Value>
            <Boolean>true</Boolean>
          </Value>
        </BACnetTimeValue>
      </ListOfTimeValues>
      <EventPriority>1252</EventPriority>
    </BACnetSpecialEvent>
  </SequenceOfBACnetSpecialEvent>
</Value>
```

SequenceOfBACnetTimeValue

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:complexType name="SequenceOfBACnetTimeValue">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetTimeValue" type="BACnetTimeValue" />
  </xs:sequence>
</xs:complexType>
```

Examples

Example 1:

```
<SequenceOfBACnetTimeValue />
```

Example 2:

```
<SequenceOfBACnetTimeValue>
  <BACnetTimeValue>
    <Time>22:39:15.980</Time>
    <Value>
      <Boolean>true</Boolean>
    </Value>
  </BACnetTimeValue>
  <BACnetTimeValue>
    <Time>22:39:15.980</Time>
    <Value>
      <Boolean>true</Boolean>
    </Value>
  </BACnetTimeValue>
  <BACnetTimeValue>
    <Time>22:39:15.980</Time>
    <Value>
      <Boolean>true</Boolean>
    </Value>
  </BACnetTimeValue>
</SequenceOfBACnetTimeValue>
```

SequenceOfBACnetTimerState

This XML representation of complex data is used by the following objects/properties:

- **timer.alarmValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetTimerState">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetTimerState"
type="BACnetTimerState" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetTimerState" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetTimerState />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetTimerState>
    <BACnetTimerState>idle</BACnetTimerState>
    <BACnetTimerState>running</BACnetTimerState>
    <BACnetTimerState>expired</BACnetTimerState>
  </SequenceOfBACnetTimerState>
</Value>
```

SequenceOfBACnetVMACEntry

This XML representation of complex data is used by the following objects/properties:

- **networkPort.virtualMacAddressTable**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetVMACEntry">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetVMACEntry"
type="BACnetVMACEntry" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetVMACEntry" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetVMACEntry />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetVMACEntry>
    <BACnetVMACEntry>
      <VirtualMacAddress>00010203040506070809</VirtualMacAddress>
      <NativeMacAddress>00010203040506070809</NativeMacAddress>
    </BACnetVMACEntry>
    <BACnetVMACEntry>
      <VirtualMacAddress>FF</VirtualMacAddress>
      <NativeMacAddress>FF</NativeMacAddress>
    </BACnetVMACEntry>
    <BACnetVMACEntry>
      <VirtualMacAddress>FF</VirtualMacAddress>
      <NativeMacAddress>FF</NativeMacAddress>
    </BACnetVMACEntry>
  </SequenceOfBACnetVMACEntry>
</Value>
```

SequenceOfBACnetVTClass

This XML representation of complex data is used by the following objects/properties:

- **device.vtClassesSupported**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetVTClass">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetVTClass" type="BACnetVTClass" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetVTClass" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetVTClass />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetVTClass>
    <BACnetVTClass>defaultTerminal</BACnetVTClass>
    <BACnetVTClass>ansiX364</BACnetVTClass>
    <BACnetVTClass>decVt52</BACnetVTClass>
  </SequenceOfBACnetVTClass>
</Value>
```

SequenceOfBACnetVTSession

This XML representation of complex data is used by the following objects/properties:

- **device.activeVtSessions**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBACnetVTSession">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BACnetVTSession"
type="BACnetVTSession" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBACnetVTSession" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetVTSession />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBACnetVTSession>
    <BACnetVTSession>
      <LocalVtSessionId>33</LocalVtSessionId>
      <RemoteVtSessionId>34</RemoteVtSessionId>
      <RemoteVtAddress>
        <NetworkNumber>7636</NetworkNumber>
        <MacAddress>00010203040506070809</MacAddress>
      </RemoteVtAddress>
    </BACnetVTSession>
    <BACnetVTSession>
      <LocalVtSessionId>36</LocalVtSessionId>
      <RemoteVtSessionId>37</RemoteVtSessionId>
      <RemoteVtAddress>
        <NetworkNumber>7639</NetworkNumber>
        <MacAddress>FF</MacAddress>
      </RemoteVtAddress>
    </BACnetVTSession>
    <BACnetVTSession>
      <LocalVtSessionId>39</LocalVtSessionId>
      <RemoteVtSessionId>40</RemoteVtSessionId>
      <RemoteVtAddress>
        <NetworkNumber>7642</NetworkNumber>
        <MacAddress>FF</MacAddress>
      </RemoteVtAddress>
    </BACnetVTSession>
  </SequenceOfBACnetVTSession>
</Value>
```

SequenceOfBitString

This XML representation of complex data is used by the following objects/properties:

- **bitstringValue.alarmValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfBitString">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="BitString" type="BitString" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfBitString" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBitString />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfBitString>
    <BitString>101</BitString>
    <BitString>1</BitString>
    <BitString>1</BitString>
  </SequenceOfBitString>
</Value>
```

SequenceOfCharacterString

This XML representation of complex data is used by the following objects/properties:

- **accessPoint.authenticationPolicyNames**
- **bitstringValue.bitText**
- **command.actionText**
- **globalGroup.groupMemberNames**
- **lift.carDoorText**
- **lift.floorText**
- **loadControl.shedLevelDescriptions**
- **multiStateInput.stateText**
- **multiStateOutput.stateText**
- **multiStateValue.stateText**
- **structuredView.subordinateAnnotations**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfCharacterString">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="CharacterString"
type="CharacterString" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfCharacterString" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfCharacterString />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfCharacterString>
    <CharacterString>Abcde</CharacterString>
    <CharacterString>Abcde</CharacterString>
    <CharacterString>Abcde</CharacterString>
  </SequenceOfCharacterString>
</Value>
```

SequenceOfOctetString

This XML representation of complex data is used by the following objects/properties:

- **networkPort.ipDnsServer**

- **networkPort.ipv6DnsServer**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfOctetString">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="OctetString" type="OctetString" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfOctetString" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfOctetString />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfOctetString>
    <OctetString>00010203040506070809</OctetString>
    <OctetString>FF</OctetString>
    <OctetString>FF</OctetString>
  </SequenceOfOctetString>
</Value>
```

SequenceOfREAL

This XML representation of complex data is used by the following objects/properties:

- **networkPort.linkSpeeds**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfREAL">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="REAL" type="REAL" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfREAL" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfREAL />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfREAL>
    <REAL>1234.567749</REAL>
    <REAL>1235.567749</REAL>
    <REAL>1236.567749</REAL>
  </SequenceOfREAL>
</Value>
```

SequenceOfReadAccessResult

This XML representation of complex data is used by the following objects/properties:

- **group.presentValue**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
  xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
  targetNamespace="http://www.elipse.com.br/drivers/BACnet"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfReadAccessResult">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="ReadAccessResult"
type="ReadAccessResult" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfReadAccessResult" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfReadAccessResult />
</Value>
```

Example 2:

```

<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfReadAccessResult>
    <ReadAccessResult>
      <ObjectIdentifier>device-501</ObjectIdentifier>
      <ListOfResults />
    </ReadAccessResult>
    <ReadAccessResult>
      <ObjectIdentifier>device-502</ObjectIdentifier>
      <ListOfResults>
        <ListOfResults_Item>
          <PropertyIdentifier>action</PropertyIdentifier>
          <PropertyArrayIndex>1237</PropertyArrayIndex>
          <ReadResult>
            <PropertyValue>
              <Boolean>true</Boolean>
            </PropertyValue>
          </ReadResult>
        </ListOfResults_Item>
      </ListOfResults>
    </ReadAccessResult>
    <ReadAccessResult>
      <ObjectIdentifier>device-505</ObjectIdentifier>
      <ListOfResults>
        <ListOfResults_Item>
          <PropertyIdentifier>activeVtSessions</PropertyIdentifier>
          <PropertyArrayIndex>1240</PropertyArrayIndex>
          <ReadResult>
            <PropertyValue>
              <BACnetVTSession>
                <LocalVtSessionId>40</LocalVtSessionId>
                <RemoteVtSessionId>41</RemoteVtSessionId>
                <RemoteVtAddress>
                  <NetworkNumber>7643</NetworkNumber>
                  <MacAddress>FF</MacAddress>
                </RemoteVtAddress>
              </BACnetVTSession>
            </PropertyValue>
          </ReadResult>
        </ListOfResults_Item>
      </ListOfResults>
    </ReadAccessResult>
  </SequenceOfReadAccessResult>
</Value>

```

Example 3:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfReadAccessResult>
    <ReadAccessResult>
      <ObjectIdentifier>device-511</ObjectIdentifier>
      <ListOfResults>
        <ListOfResults_Item>
          <PropertyIdentifier>apduTimeout</PropertyIdentifier>
          <PropertyArrayIndex>1246</PropertyArrayIndex>
          <ReadResult>
            <PropertyValue>
              <Unsigned>1247</Unsigned>
            </PropertyValue>
          </ReadResult>
        </ListOfResults_Item>
        <ListOfResults_Item>
          <PropertyIdentifier>bias</PropertyIdentifier>
          <PropertyArrayIndex>1249</PropertyArrayIndex>
          <ReadResult>
            <PropertyValue>
              <Boolean>true</Boolean>
            </PropertyValue>
          </ReadResult>
        </ListOfResults_Item>
        <ListOfResults_Item>
          <PropertyIdentifier>changeOfStateTime</PropertyIdentifier>
          <PropertyArrayIndex>1251</PropertyArrayIndex>
          <ReadResult>
            <PropertyValue>
              <Boolean>true</Boolean>
            </PropertyValue>
          </ReadResult>
        </ListOfResults_Item>
      </ListOfResults>
    </ReadAccessResult>
    <ReadAccessResult>
      <ObjectIdentifier>device-519</ObjectIdentifier>
      <ListOfResults>
        <ListOfResults_Item>
          <PropertyIdentifier>controlledVariableUnits</PropertyIdentifier>
          <PropertyArrayIndex>1254</PropertyArrayIndex>
          <ReadResult>
            <PropertyValue>
              <Boolean>true</Boolean>
            </PropertyValue>
          </ReadResult>
        </ListOfResults_Item>
      </ListOfResults>
    </ReadAccessResult>
    <ReadAccessResult>
      <ObjectIdentifier>device-522</ObjectIdentifier>
      <ListOfResults>
        <ListOfResults_Item>
          <PropertyIdentifier>dateList</PropertyIdentifier>
          <PropertyArrayIndex>1257</PropertyArrayIndex>
          <ReadResult>
            <PropertyValue>
              <Boolean>true</Boolean>
            </PropertyValue>
          </ReadResult>
        </ListOfResults_Item>
      </ListOfResults>
    </ReadAccessResult>
  </SequenceOfReadAccessResult>
</Value>
```

SequenceOfReadAccessSpecification

This XML representation of complex data is used by the following objects/properties:

- `group.listOfGroupMembers`

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfReadAccessSpecification">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="ReadAccessSpecification"
type="ReadAccessSpecification" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfReadAccessSpecification" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfReadAccessSpecification />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfReadAccessSpecification>
    <ReadAccessSpecification>
      <ObjectIdentifier>device-501</ObjectIdentifier>
      <ListOfPropertyReferences />
    </ReadAccessSpecification>
    <ReadAccessSpecification>
      <ObjectIdentifier>device-502</ObjectIdentifier>
      <ListOfPropertyReferences>
        <BACnetPropertyReference>
          <PropertyIdentifier>action</PropertyIdentifier>
          <PropertyArrayIndex>1237</PropertyArrayIndex>
        </BACnetPropertyReference>
      </ListOfPropertyReferences>
    </ReadAccessSpecification>
    <ReadAccessSpecification>
      <ObjectIdentifier>device-505</ObjectIdentifier>
      <ListOfPropertyReferences>
        <BACnetPropertyReference>
          <PropertyIdentifier>activeVtSessions</PropertyIdentifier>
          <PropertyArrayIndex>1240</PropertyArrayIndex>
        </BACnetPropertyReference>
      </ListOfPropertyReferences>
    </ReadAccessSpecification>
  </SequenceOfReadAccessSpecification>
</Value>
```

Example 3:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <SequenceOfReadAccessSpecification>
    <ReadAccessSpecification>
      <ObjectIdentifier>device-508</ObjectIdentifier>
      <ListOfPropertyReferences>
        <BACnetPropertyReference>
          <PropertyIdentifier>all</PropertyIdentifier>
          <PropertyArrayIndex>1243</PropertyArrayIndex>
        </BACnetPropertyReference>
        <BACnetPropertyReference>
          <PropertyIdentifier>apduSegmentTimeout</PropertyIdentifier>
          <PropertyArrayIndex>1245</PropertyArrayIndex>
        </BACnetPropertyReference>
        <BACnetPropertyReference>
          <PropertyIdentifier>applicationSoftwareVersion</PropertyIdentifier>
          <PropertyArrayIndex>1247</PropertyArrayIndex>
        </BACnetPropertyReference>
      </ListOfPropertyReferences>
    </ReadAccessSpecification>
    <ReadAccessSpecification>
      <ObjectIdentifier>device-515</ObjectIdentifier>
      <ListOfPropertyReferences>
        <BACnetPropertyReference>
          <PropertyIdentifier>changeOfStateCount</PropertyIdentifier>
          <PropertyArrayIndex>1250</PropertyArrayIndex>
        </BACnetPropertyReference>
      </ListOfPropertyReferences>
    </ReadAccessSpecification>
    <ReadAccessSpecification>
      <ObjectIdentifier>device-518</ObjectIdentifier>
      <ListOfPropertyReferences>
        <BACnetPropertyReference>
          <PropertyIdentifier>controlledVariableReference</PropertyIdentifier>
          <PropertyArrayIndex>1253</PropertyArrayIndex>
        </BACnetPropertyReference>
      </ListOfPropertyReferences>
    </ReadAccessSpecification>
  </SequenceOfReadAccessSpecification>
</Value>
```

SequenceOfUnsigned

This XML representation of complex data is used by the following objects/properties:

- **channel.executionDelay**
- **credentialDataInput.supportedFormatClasses**
- **loadControl.shedLevels**
- **multiStateInput.alarmValues**
- **multiStateInput.faultValues**
- **multiStateValue.alarmValues**
- **multiStateValue.faultValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfUnsigned">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Unsigned" type="Unsigned" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfUnsigned" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfUnsigned />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfUnsigned>
    <Unsigned>1234</Unsigned>
    <Unsigned>1235</Unsigned>
    <Unsigned>1236</Unsigned>
  </SequenceOfUnsigned>
</Value>
```

SequenceOfUnsigned32

This XML representation of complex data is used by the following objects/properties:

- **channel.controlGroups**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfUnsigned32">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Unsigned32" type="Unsigned32" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfUnsigned32" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfUnsigned32 />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfUnsigned32>
    <Unsigned32>2345</Unsigned32>
    <Unsigned32>2346</Unsigned32>
    <Unsigned32>2347</Unsigned32>
  </SequenceOfUnsigned32>
</Value>
```

SequenceOfUnsigned8

This XML representation of complex data is used by the following objects/properties:

- **lift.makingCarCall**
- **networkSecurity.supportedSecurityAlgorithms**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="SequenceOfUnsigned8">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="Unsigned8" type="Unsigned8" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Value" type="SequenceOfUnsigned8" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfUnsigned8 />
</Value>
```

Example 2:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <SequenceOfUnsigned8>
    <Unsigned8>33</Unsigned8>
    <Unsigned8>34</Unsigned8>
    <Unsigned8>35</Unsigned8>
  </SequenceOfUnsigned8>
</Value>
```

Time

This XML representation of complex data is used by the following objects/properties:

- **timer.stateChangeValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="Time">
    <xs:restriction base="xs:string">
      <xs:pattern value="((([01][0-9]|2[0-3])|\?\?):([0-5][0-9]|\?\?):([0-5][0-9]|\?\?)(.([0-9][0-9]
[0-9]|\?\?\?))?)?" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="Value" type="Time" />
</xs:schema>
```

Examples

Example 1:


```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <Time>22:39:15.980</Time>
</Value>
```

Unsigned

This XML representation of complex data is used by the following objects/properties:

- **accessDoor.priorityArray**
- **analogOutput.priorityArray**
- **analogValue.priorityArray**
- **binaryLightingOutput.priorityArray**
- **binaryOutput.priorityArray**
- **binaryValue.priorityArray**
- **bitstringValue.priorityArray**
- **channel.presentValue**
- **characterstringValue.priorityArray**
- **dateValue.priorityArray**
- **datepatternValue.priorityArray**
- **datetimeValue.priorityArray**
- **datetimepatternValue.priorityArray**
- **integerValue.priorityArray**
- **largeAnalogValue.priorityArray**
- **lightingOutput.priorityArray**
- **loadControl.actualShedLevel**
- **loadControl.expectedShedLevel**
- **loadControl.requestedShedLevel**
- **multiStateOutput.priorityArray**
- **multiStateValue.priorityArray**
- **octetstringValue.priorityArray**
- **positiveIntegerValue.priorityArray**
- **timeValue.priorityArray**
- **timepatternValue.priorityArray**
- **timer.stateChangeValues**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.elipse.com.br/drivers/BACnet"
xmlns:msns="http://www.elipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.elipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="Unsigned">
    <xs:restriction base="xs:unsignedInt" />
  </xs:simpleType>
  <xs:element name="Value" type="Unsigned" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.elipse.com.br/drivers/BACnet">
  <Unsigned>1234</Unsigned>
</Value>
```

Unsigned16

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="Unsigned16">
  <xs:restriction base="xs:unsignedShort" />
</xs:simpleType>
```

Examples

Example 1:

```
<Unsigned16>7634</Unsigned16>
```

Unsigned32

This XML representation of complex data is used internally by other XMLs supported by the driver.

XML Schema

```
<xs:simpleType name="Unsigned32">
  <xs:restriction base="xs:unsignedInt" />
</xs:simpleType>
```

Examples

Example 1:

```
<Unsigned32>2345</Unsigned32>
```

Unsigned8

This XML representation of complex data is used by the following objects/properties:

- **elevatorGroup.landingCallControl**

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://www.eclipse.com.br/drivers/BACnet"
xmlns:mstns="http://www.eclipse.com.br/drivers/BACnet" elementFormDefault="qualified"
targetNamespace="http://www.eclipse.com.br/drivers/BACnet"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="Unsigned8">
    <xs:restriction base="xs:unsignedByte" />
  </xs:simpleType>
  <xs:element name="Value" type="Unsigned8" />
</xs:schema>
```

Examples

Example 1:

```
<?xml version="1.0" encoding="utf-8" ?>
<Value xmlns="http://www.eclipse.com.br/drivers/BACnet">
  <Unsigned8>33</Unsigned8>
</Value>
```

BACnet Enumerations

The following BACnet enumerations are supported by the driver:

- **BACnetAccessCredentialDisable**
- **BACnetAccessEvent**
- **BACnetAccessPassbackMode**
- **BACnetAccessUserType**
- **BACnetAccessZoneOccupancyState**
- **BACnetAccumulatorRecord_AccumulatorStatus**
- **BACnetAction**
- **BACnetAuthenticationFactorType**
- **BACnetAuthenticationStatus**
- **BACnetAuthorizationMode**
- **BACnetBackupState**
- **BACnetBinaryLightingPV**
- **BACnetBinaryPV**
- **BACnetDeviceStatus**
- **BACnetDoorAlarmState**
- **BACnetDoorSecuredStatus**
- **BACnetDoorStatus**
- **BACnetDoorValue**
- **BACnetEngineeringUnits**
- **BACnetEscalatorMode**
- **BACnetEscalatorOperationDirection**

- **BACnetEventState**
- **BACnetEventType**
- **BACnetFaultType**
- **BACnetFileAccessMethod**
- **BACnetIPMode**
- **BACnetLifeSafetyMode**
- **BACnetLifeSafetyOperation**
- **BACnetLifeSafetyState**
- **BACnetLiftCarDirection**
- **BACnetLiftCarDriveStatus**
- **BACnetLiftCarMode**
- **BACnetLiftGroupMode**
- **BACnetLightingInProgress**
- **BACnetLightingOperation**
- **BACnetLightingTransition**
- **BACnetLockStatus**
- **BACnetLoggingType**
- **BACnetMaintenance**
- **BACnetNetworkNumberQuality**
- **BACnetNetworkPortCommand**
- **BACnetNetworkType**
- **BACnetNodeType**
- **BACnetNotifyType**
- **BACnetObjectType**
- **BACnetPolarity**
- **BACnetProgramError**
- **BACnetProgramRequest**
- **BACnetProgramState**
- **BACnetPropertyIdentifier**
- **BACnetProtocolLevel**
- **BACnetRelationship**
- **BACnetReliability**

- **BACnetRestartReason**
- **BACnetSecurityLevel**
- **BACnetSegmentation**
- **BACnetShedState**
- **BACnetSilencedState**
- **BACnetTimerState**
- **BACnetTimerTransition**
- **BACnetWriteStatus**

BACnetAccessCredentialDisable

This enumeration is used by the following objects/properties:

- **accessCredential.credentialDisable**

The enumeration values are:

Value	Name
0	none
1	disable
2	disableManual
3	disableLockout

BACnetAccessEvent

This enumeration is used by the following objects/properties:

- **accessCredential.lastAccessEvent**
- **accessPoint.accessEvent**

The enumeration values are:

Value	Name
0	none
1	granted
2	muster
3	passbackDetected
4	duress
5	trace
6	lockoutMaxAttempts
7	lockoutOther
8	lockoutRelinquished
9	lockedByHigherPriority
10	outOfService

11	outOfServiceRelinquished
12	accompanimentBy
13	authenticationFactorRead
14	authorizationDelayed
15	verificationRequired
16	noEntryAfterGranted
128	deniedDenyAll
129	deniedUnknownCredential
130	deniedAuthenticationUnavailable
131	deniedAuthenticationFactorTimeout
132	deniedIncorrectAuthenticationFactor
133	deniedZoneNoAccessRights
134	deniedPointNoAccessRights
135	deniedNoAccessRights
136	deniedOutOfTimeRange
137	deniedThreatLevel
138	deniedPassback
139	deniedUnexpectedLocationUsage
140	deniedMaxAttempts
141	deniedLowerOccupancyLimit
142	deniedUpperOccupancyLimit
143	deniedAuthenticationFactorLost
144	deniedAuthenticationFactorStolen
145	deniedAuthenticationFactorDamaged
146	deniedAuthenticationFactorDestroyed
147	deniedAuthenticationFactorDisabled
148	deniedAuthenticationFactorError
149	deniedCredentialUnassigned
150	deniedCredentialNotProvisioned
151	deniedCredentialNotYetActive
152	deniedCredentialExpired
153	deniedCredentialManualDisable
154	deniedCredentialLockout
155	deniedCredentialMaxDays
156	deniedCredentialMaxUses
157	deniedCredentialInactivity
158	deniedCredentialDisabled
159	deniedNoAccompaniment
160	deniedIncorrectAccompaniment

161	deniedLockout
162	deniedVerificationFailed
163	deniedVerificationTimeout
164	deniedOther

BACnetAccessPassbackMode

This enumeration is used by the following objects/properties:

- **accessZone.passbackMode**

The enumeration values are:

Value	Name
0	passbackOff
1	hardPassback
2	softPassback

BACnetAccessUserType

This enumeration is used by the following objects/properties:

- **accessUser.userType**

The enumeration values are:

Value	Name
0	asset
1	group
2	person

BACnetAccessZoneOccupancyState

This enumeration is used by the following objects/properties:

- **accessZone.occupancyState**

The enumeration values are:

Value	Name
0	normal
1	belowLowerLimit
2	atLowerLimit
3	atUpperLimit
4	aboveUpperLimit
5	disabled
6	notSupported

BACnetAccumulatorRecord_AccumulatorStatus

This enumeration is used by the following objects/properties:

- **accumulator.loggingRecord**

The enumeration values are:

Value	Name
0	normal
1	starting
2	recovered
3	abnormal
4	failed

BACnetAction

This enumeration is used by the following objects/properties:

- **loop.action**

The enumeration values are:

Value	Name
0	direct
1	reverse

BACnetAuthenticationFactorType

This enumeration is used by the following objects/properties:

- **accessPoint.accessEventAuthenticationFactor**
- **credentialDataInput.presentValue**

The enumeration values are:

Value	Name
0	undefined
1	error
2	custom
3	simpleNumber16
4	simpleNumber32
5	simpleNumber56
6	simpleAlphaNumeric
7	abaTrack2
8	wiegand26
9	wiegand37
10	wiegand37Facility
11	facility16Card32
12	facility32Card32
13	fascN
14	fascNBcd
15	fascNLarge
16	fascNLargeBcd
17	gsa75
18	chuid
19	chuidFull
20	guid
21	cbeffA
22	cbeffB
23	cbeffC
24	userPassword

BACnetAuthenticationStatus

This enumeration is used by the following objects/properties:

- **accessPoint.authenticationStatus**

The enumeration values are:

Value	Name
0	notReady
1	ready
2	disabled
3	waitingForAuthenticationFactor
4	waitingForAccompaniment
5	waitingForVerification
6	inProgress

BACnetAuthorizationMode

This enumeration is used by the following objects/properties:

- **accessPoint.authorizationMode**

The enumeration values are:

Value	Name
0	authorize
1	grantActive
2	denyAll
3	verificationRequired
4	authorizationDelayed
5	none

BACnetBackupState

This enumeration is used by the following objects/properties:

- **device.backupAndRestoreState**

The enumeration values are:

Value	Name
0	idle
1	preparingForBackup
2	preparingForRestore
3	performingABackup
4	performingARestore
5	backupFailure
6	restoreFailure

BACnetBinaryLightingPV

This enumeration is used by the following objects/properties:

- **binaryLightingOutput.feedbackValue**

- **binaryLightingOutput.presentValue**
- **binaryLightingOutput.relinquishDefault**

The enumeration values are:

Value	Name
0	off
1	on
2	warn
3	warnOff
4	warnRelinquish
5	stop

BACnetBinaryPV

This enumeration is used by the following objects/properties:

- **accessCredential.credentialStatus**
- **binaryInput.alarmValue**
- **binaryInput.interfaceValue**
- **binaryInput.presentValue**
- **binaryOutput.feedbackValue**
- **binaryOutput.interfaceValue**
- **binaryOutput.presentValue**
- **binaryOutput.relinquishDefault**
- **binaryValue.alarmValue**
- **binaryValue.presentValue**
- **binaryValue.relinquishDefault**

The enumeration values are:

Value	Name
0	inactive
1	active

BACnetDeviceStatus

This enumeration is used by the following objects/properties:

- **device.systemStatus**

The enumeration values are:

Value	Name
0	operational
1	operationalReadOnly
2	downloadRequired
3	downloadInProgress
4	nonOperational
5	backupInProgress

BACnetDoorAlarmState

This enumeration is used by the following objects/properties:

- **accessDoor.doorAlarmState**

The enumeration values are:

Value	Name
0	normal
1	alarm
2	doorOpenTooLong
3	forcedOpen
4	tamper
5	doorFault
6	lockDown
7	freeAccess
8	egressOpen

BACnetDoorSecuredStatus

This enumeration is used by the following objects/properties:

- **accessDoor.securedStatus**

The enumeration values are:

Value	Name
0	secured
1	unsecured
2	unknown

BACnetDoorStatus

This enumeration is used by the following objects/properties:

- **accessDoor.doorStatus**

The enumeration values are:

Value	Name
0	closed
1	opened
2	unknown
3	doorFault
4	unused
5	none
6	closing
7	opening
8	safetyLocked
9	limitedOpened

BACnetDoorValue

This enumeration is used by the following objects/properties:

- **accessDoor.presentValue**
- **accessDoor.relinquishDefault**

The enumeration values are:

Value	Name
0	lock
1	unlock
2	pulseUnlock
3	extendedPulseUnlock

BACnetEngineeringUnits

This enumeration is used by the following objects/properties:

- **accumulator.units**
- **analogInput.units**
- **analogOutput.units**
- **analogValue.units**
- **integerValue.units**
- **largeAnalogValue.units**
- **lifeSafetyPoint.units**
- **lift.carLoadUnits**
- **loop.controlledVariableUnits**
- **loop.derivativeConstantUnits**

- **loop.integralConstantUnits**
- **loop.outputUnits**
- **loop.proportionalConstantUnits**
- **positiveIntegerValue.units**
- **pulseConverter.units**

The enumeration values are:

Value	Name
0	squareMeters
1	squareFeet
2	milliamperes
3	amperes
4	ohms
5	volts
6	kilovolts
7	megavolts
8	voltAmperes
9	kilovoltAmperes
10	megavoltAmperes
11	voltAmperesReactive
12	kilovoltAmperesReactive
13	megavoltAmperesReactive
14	degreesPhase
15	powerFactor
16	joules
17	kilojoules
18	wattHours
19	kilowattHours
20	btus
21	therms
22	tonHours
23	joulesPerKilogramDryAir
24	btusPerPoundDryAir
25	cyclesPerHour
26	cyclesPerMinute
27	hertz
28	gramsOfWaterPerKilogramDryAir
29	percentRelativeHumidity
30	millimeters
31	meters
32	inches
33	feet
34	wattsPerSquareFoot
35	wattsPerSquareMeter
36	lumens
37	luxes
38	footCandles
39	kilograms
40	poundsMass

BACnetEscalatorMode

This enumeration is used by the following objects/properties:

- **escalator.escalatorMode**

The enumeration values are:

Value	Name
0	unknown
1	stop
2	up
3	down
4	inspection
5	outOfService

BACnetEscalatorOperationDirection

This enumeration is used by the following objects/properties:

- **escalator.operationDirection**

The enumeration values are:

Value	Name
0	unknown
1	stopped
2	upRatedSpeed
3	upReducedSpeed
4	downRatedSpeed
5	downReducedSpeed

BACnetEventState

This enumeration is used by the following objects/properties:

- **accessDoor.eventState**
- **accessPoint.eventState**
- **accessZone.eventState**
- **accumulator.eventState**
- **alertEnrollment.eventState**
- **analogInput.eventState**
- **analogOutput.eventState**
- **analogValue.eventState**
- **binaryInput.eventState**

- **binaryOutput.eventState**
- **binaryValue.eventState**
- **bitstringValue.eventState**
- **channel.eventState**
- **characterstringValue.eventState**
- **command.eventState**
- **credentialDataInput.eventState**
- **dateValue.eventState**
- **datepatternValue.eventState**
- **datetimeValue.eventState**
- **datetimepatternValue.eventState**
- **device.eventState**
- **escalator.eventState**
- **eventEnrollment.eventState**
- **eventLog.eventState**
- **globalGroup.eventState**
- **integerValue.eventState**
- **largeAnalogValue.eventState**
- **lifeSafetyPoint.eventState**
- **lifeSafetyZone.eventState**
- **lift.eventState**
- **loadControl.eventState**
- **loop.eventState**
- **multiStateInput.eventState**
- **multiStateOutput.eventState**
- **multiStateValue.eventState**
- **networkPort.eventState**
- **notificationClass.eventState**
- **octetstringValue.eventState**
- **positiveIntegerValue.eventState**
- **program.eventState**
- **pulseConverter.eventState**

- **schedule.eventState**
- **timeValue.eventState**
- **timepatternValue.eventState**
- **timer.eventState**
- **trendLog.eventState**
- **trendLogMultiple.eventState**

The enumeration values are:

Value	Name
0	normal
1	fault
2	offnormal
3	highLimit
4	lowLimit
5	lifeSafetyAlarm

BACnetEventType

This enumeration is used by the following objects/properties:

- **eventEnrollment.eventType**

The enumeration values are:

Value	Name
0	changeOfBitstring
1	changeOfState
2	changeOfValue
3	commandFailure
4	floatingLimit
5	outOfRange
8	changeOfLifeSafety
9	extended
10	bufferReady
11	unsignedRange
13	accessEvent
14	doubleOutOfRange
15	signedOutOfRange
16	unsignedOutOfRange
17	changeOfCharacterstring
18	changeOfStatusFlags
19	changeOfReliability
20	none
21	changeOfDiscreteValue
22	changeOfTimer

BACnetFaultType

This enumeration is used by the following objects/properties:

- **eventEnrollment.faultType**

The enumeration values are:

Value	Name
0	none
1	faultCharacterstring
2	faultExtended
3	faultLifeSafety
4	faultState
5	faultStatusFlags
6	faultOutOfRange
7	faultListed

BACnetFileAccessMethod

This enumeration is used by the following objects/properties:

- **file.fileAccessMethod**

The enumeration values are:

Value	Name
0	recordAccess
1	streamAccess

BACnetIPMode

This enumeration is used by the following objects/properties:

- **networkPort.bacnetIpMode**
- **networkPort.bacnetIpv6Mode**

The enumeration values are:

Value	Name
0	normal
1	foreign
2	bbmd

BACnetLifeSafetyMode

This enumeration is used by the following objects/properties:

- **lifeSafetyPoint.mode**
- **lifeSafetyZone.mode**

The enumeration values are:

Value	Name
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	automaticReleaseDisabled
14	default

BACnetLifeSafetyOperation

This enumeration is used by the following objects/properties:

- **lifeSafetyPoint.operationExpected**
- **lifeSafetyZone.operationExpected**

The enumeration values are:

Value	Name
0	none
1	silence
2	silenceAudible
3	silenceVisual
4	reset
5	resetAlarm
6	resetFault
7	unsilence
8	unsilenceAudible
9	unsilenceVisual

BACnetLifeSafetyState

This enumeration is used by the following objects/properties:

- **lifeSafetyPoint.presentValue**
- **lifeSafetyPoint.trackingValue**

- **lifeSafetyZone.presentValue**
- **lifeSafetyZone.trackingValue**

The enumeration values are:

Value	Name
0	quiet
1	preAlarm
2	alarm
3	fault
4	faultPreAlarm
5	faultAlarm
6	notReady
7	active
8	tamper
9	testAlarm
10	testActive
11	testFault
12	testFaultAlarm
13	holdup
14	duress
15	tamperAlarm
16	abnormal
17	emergencyPower
18	delayed
19	blocked
20	localAlarm
21	generalAlarm
22	supervisory
23	testSupervisory

BACnetLiftCarDirection

This enumeration is used by the following objects/properties:

- **elevatorGroup.landingCallControl**
- **lift.carAssignedDirection**
- **lift.carMovingDirection**

The enumeration values are:

Value	Name
0	unknown
1	none
2	stopped
3	up
4	down
5	upAndDown

BACnetLiftCarDriveStatus

This enumeration is used by the following objects/properties:

- **lift.carDriveStatus**

The enumeration values are:

Value	Name
0	unknown
1	stationary
2	braking
3	accelerate
4	decelerate
5	ratedSpeed
6	singleFloorJump
7	twoFloorJump
8	threeFloorJump
9	multiFloorJump

BACnetLiftCarMode

This enumeration is used by the following objects/properties:

- **lift.carMode**

The enumeration values are:

Value	Name
0	unknown
1	normal
2	vip
3	homing
4	parking
5	attendantControl
6	firefighterControl
7	emergencyPower
8	inspection
9	cabinetRecall
10	earthquakeOperation
11	fireOperation
12	outOfService
13	occupantEvacuation

BACnetLiftGroupMode

This enumeration is used by the following objects/properties:

- **elevatorGroup.groupMode**

The enumeration values are:

Value	Name
0	unknown
1	normal
2	downPeak
3	twoWay
4	fourWay
5	emergencyPower
6	upPeak

BACnetLightingInProgress

This enumeration is used by the following objects/properties:

- **lightingOutput.inProgress**

The enumeration values are:

Value	Name
0	idle
1	fadeActive
2	rampActive
3	notControlled
4	other

BACnetLightingOperation

This enumeration is used by the following objects/properties:

- **lightingOutput.lightingCommand**

The enumeration values are:

Value	Name
0	none
1	fadeTo
2	rampTo
3	stepUp
4	stepDown
5	stepOn
6	stepOff
7	warn
8	warnOff
9	warnRelinquish
10	stop

BACnetLightingTransition

This enumeration is used by the following objects/properties:

- **lightingOutput.transition**

The enumeration values are:

Value	Name
0	none
1	fade
2	ramp

BACnetLockStatus

This enumeration is used by the following objects/properties:

- **accessDoor.lockStatus**

The enumeration values are:

Value	Name
0	locked
1	unlocked
2	lockFault
3	unused
4	unknown

BACnetLoggingType

This enumeration is used by the following objects/properties:

- **trendLog.loggingType**
- **trendLogMultiple.loggingType**

The enumeration values are:

Value	Name
0	polled
1	cov
2	triggered

BACnetMaintenance

This enumeration is used by the following objects/properties:

- **accessDoor.maintenanceRequired**
- **lifeSafetyPoint.maintenanceRequired**

The enumeration values are:

Value	Name
0	none
1	periodicTest
2	needServiceOperational
3	needServiceInoperative

BACnetNetworkNumberQuality

This enumeration is used by the following objects/properties:

- **networkPort.networkNumberQuality**

The enumeration values are:

Value	Name
0	unknown
1	learned
2	learnedConfigured
3	configured

BACnetNetworkPortCommand

This enumeration is used by the following objects/properties:

- **networkPort.command**

The enumeration values are:

Value	Name
0	idle
1	discardChanges
2	renewFdRegistration
3	restartSlaveDiscovery
4	renewDhcp
5	restartAutonegotiation
6	disconnect
7	restartPort

BACnetNetworkType

This enumeration is used by the following objects/properties:

- **networkPort.networkType**

The enumeration values are:

Value	Name
0	ethernet
1	arcnet
2	mstp
3	ptp
4	lontalk
5	ipv4
6	zigbee
7	virtual
9	ipv6
10	serial

BACnetNodeType

This enumeration is used by the following objects/properties:

- **structuredView.nodeType**

The enumeration values are:

Value	Name
0	unknown
1	system
2	network
3	device
4	organizational
5	area
6	equipment
7	point
8	collection
9	property
10	functional
11	other
12	subsystem
13	building
14	floor
15	section
16	module
17	tree
18	member
19	protocol
20	room
21	zone

BACnetNotifyType

This enumeration is used by the following objects/properties:

- **accessDoor.notifyType**
- **accessPoint.notifyType**
- **accessZone.notifyType**
- **accumulator.notifyType**
- **alertEnrollment.notifyType**
- **analogInput.notifyType**

- **analogOutput.notifyType**
- **analogValue.notifyType**
- **binaryInput.notifyType**
- **binaryLightingOutput.notifyType**
- **binaryOutput.notifyType**
- **binaryValue.notifyType**
- **bitstringValue.notifyType**
- **channel.notifyType**
- **characterstringValue.notifyType**
- **command.notifyType**
- **credentialDataInput.notifyType**
- **dateValue.notifyType**
- **datepatternValue.notifyType**
- **datetimeValue.notifyType**
- **datetimepatternValue.notifyType**
- **device.notifyType**
- **escalator.notifyType**
- **eventEnrollment.notifyType**
- **eventLog.notifyType**
- **globalGroup.notifyType**
- **integerValue.notifyType**
- **largeAnalogValue.notifyType**
- **lifeSafetyPoint.notifyType**
- **lifeSafetyZone.notifyType**
- **lift.notifyType**
- **loadControl.notifyType**
- **loop.notifyType**
- **multiStateInput.notifyType**
- **multiStateOutput.notifyType**
- **multiStateValue.notifyType**
- **networkPort.notifyType**
- **notificationClass.notifyType**

- **positiveIntegerValue.notifyType**
- **program.notifyType**
- **pulseConverter.notifyType**
- **schedule.notifyType**
- **timeValue.notifyType**
- **timepatternValue.notifyType**
- **timer.notifyType**
- **trendLog.notifyType**
- **trendLogMultiple.notifyType**

The enumeration values are:

Value	Name
0	alarm
1	event
2	ackNotification

BACnetObjectType

This enumeration is used by the following objects/properties:

- **accessCredential.objectType**
- **accessDoor.objectType**
- **accessPoint.objectType**
- **accessRights.objectType**
- **accessUser.objectType**
- **accessZone.objectType**
- **accumulator.objectType**
- **alertEnrollment.objectType**
- **analogInput.objectType**
- **analogOutput.objectType**
- **analogValue.objectType**
- **averaging.objectType**
- **binaryInput.objectType**
- **binaryLightingOutput.objectType**
- **binaryOutput.objectType**
- **binaryValue.objectType**

- **bitstringValue.objectType**
- **calendar.objectType**
- **channel.objectType**
- **characterstringValue.objectType**
- **command.objectType**
- **credentialDataInput.objectType**
- **dateValue.objectType**
- **datepatternValue.objectType**
- **datetimeValue.objectType**
- **datetimepatternValue.objectType**
- **device.objectType**
- **elevatorGroup.objectType**
- **escalator.objectType**
- **eventEnrollment.objectType**
- **eventLog.objectType**
- **file.objectType**
- **globalGroup.objectType**
- **group.objectType**
- **integerValue.objectType**
- **largeAnalogValue.objectType**
- **lifeSafetyPoint.objectType**
- **lifeSafetyZone.objectType**
- **lift.objectType**
- **lightingOutput.objectType**
- **loadControl.objectType**
- **loop.objectType**
- **multiStateInput.objectType**
- **multiStateOutput.objectType**
- **multiStateValue.objectType**
- **networkPort.objectType**
- **networkSecurity.objectType**
- **notificationClass.objectType**

- **notificationForwarder.objectType**
- **octetstringValue.objectType**
- **positiveIntegerValue.objectType**
- **program.objectType**
- **pulseConverter.objectType**
- **schedule.objectType**
- **structuredView.objectType**
- **timeValue.objectType**
- **timepatternValue.objectType**
- **timer.objectType**
- **trendLog.objectType**
- **trendLogMultiple.objectType**

The enumeration values are:

Value	Name
0	analogInput
1	analogOutput
2	analogValue
3	binaryInput
4	binaryOutput
5	binaryValue
6	calendar
7	command
8	device
9	eventEnrollment
10	file
11	group
12	loop
13	multiStateInput
14	multiStateOutput
15	notificationClass
16	program
17	schedule
18	averaging
19	multiStateValue
20	trendLog
21	lifeSafetyPoint
22	lifeSafetyZone
23	accumulator
24	pulseConverter
25	eventLog
26	globalGroup
27	trendLogMultiple
28	loadControl
29	structuredView
30	accessDoor
31	timer
32	accessCredential
33	accessPoint
34	accessRights
35	accessUser
36	accessZone
37	credentialDataInput
38	networkSecurity
39	bitstringValue
40	characterstringValue

BACnetPolarity

This enumeration is used by the following objects/properties:

- **binaryInput.polarity**
- **binaryLightingOutput.polarity**
- **binaryOutput.polarity**

The enumeration values are:

Value	Name
0	normal
1	reverse

BACnetProgramError

This enumeration is used by the following objects/properties:

- **program.reasonForHalt**

The enumeration values are:

Value	Name
0	normal
1	loadFailed
2	internal
3	program
4	other

BACnetProgramRequest

This enumeration is used by the following objects/properties:

- **program.programChange**

The enumeration values are:

Value	Name
0	ready
1	load
2	run
3	halt
4	restart
5	unload

BACnetProgramState

This enumeration is used by the following objects/properties:

- **program.programState**

The enumeration values are:

Value	Name
0	idle
1	loading
2	running
3	waiting
4	halted
5	unloading

BACnetPropertyIdentifier

This enumeration is used by the following objects/properties:

- **averaging.objectPropertyReference**
- **eventEnrollment.objectPropertyReference**
- **trendLog.logDeviceObjectProperty**

The enumeration values are:

Value	Name
0	ackedTransitions
1	ackRequired
2	action
3	actionText
4	activeText
5	activeVtSessions
6	alarmValue
7	alarmValues
8	all
9	allWritesSuccessful
10	apduSegmentTimeout
11	apduTimeout
12	applicationSoftwareVersion
13	archive
14	bias
15	changeOfStateCount
16	changeOfStateTime
17	notificationClass
19	controlledVariableReference
20	controlledVariableUnits

21	controlledVariableValue
22	covIncrement
23	dateList
24	daylightSavingsStatus
25	deadband
26	derivativeConstant
27	derivativeConstantUnits
28	description
29	descriptionOfHalt
30	deviceAddressBinding
31	deviceType
32	effectivePeriod
33	elapsedActiveTime
34	errorLimit
35	eventEnable
36	eventState
37	eventType
38	exceptionSchedule
39	faultValues
40	feedbackValue
41	fileAccessMethod
42	fileSize
43	fileType
44	firmwareRevision
45	highLimit
46	inactiveText
47	inProcess
48	instanceOf
49	integralConstant
50	integralConstantUnits
52	limitEnable
53	listOfGroupMembers
54	listOfObjectPropertyReferences
56	localDate
57	localTime
58	location
59	lowLimit
60	manipulatedVariableReference
61	maximumOutput

62	maxAduLengthAccepted
63	maxInfoFrames
64	maxMaster
65	maxPresValue
66	minimumOffTime
67	minimumOnTime
68	minimumOutput
69	minPresValue
70	modelName
71	modificationDate
72	notifyType
73	numberOfAduRetries
74	numberOfStates
75	objectIdentifier
76	objectList
77	objectName
78	objectPropertyReference
79	objectType
80	optional
81	outOfService
82	outputUnits
83	eventParameters
84	polarity
85	presentValue
86	priority
87	priorityArray
88	priorityForWriting
89	processIdentifier
90	programChange
91	programLocation
92	programState
93	proportionalConstant
94	proportionalConstantUnits
96	protocolObjectTypesSupported
97	protocolServicesSupported
98	protocolVersion
99	readOnly
100	reasonForHalt
102	recipientList

103	reliability
104	relinquishDefault
105	required
106	resolution
107	segmentationSupported
108	setpoint
109	setpointReference
110	stateText
111	statusFlags
112	systemStatus
113	timeDelay
114	timeOfActiveTimeReset
115	timeOfStateCountReset
116	timeSynchronizationRecipients
117	units
118	updateInterval
119	utcOffset
120	vendorIdentifier
121	vendorName
122	vtClassesSupported
123	weeklySchedule
124	attemptedSamples
125	averageValue
126	bufferSize
127	clientCovIncrement
128	covResubscriptionInterval
130	eventTimeStamps
131	logBuffer
132	logDeviceObjectProperty
133	enable
134	logInterval
135	maximumValue
136	minimumValue
137	notificationThreshold
139	protocolRevision
140	recordsSinceNotification
141	recordCount
142	startTime
143	stopTime

144	stopWhenFull
145	totalRecordCount
146	validSamples
147	windowInterval
148	windowSamples
149	maximumValueTimestamp
150	minimumValueTimestamp
151	varianceValue
152	activeCovSubscriptions
153	backupFailureTimeout
154	configurationFiles
155	databaseRevision
156	directReading
157	lastRestoreTime
158	maintenanceRequired
159	memberOf
160	mode
161	operationExpected
162	setting
163	silenced
164	trackingValue
165	zoneMembers
166	lifeSafetyAlarmValues
167	maxSegmentsAccepted
168	profileName
169	autoSlaveDiscovery
170	manualSlaveAddressBinding
171	slaveAddressBinding
172	slaveProxyEnable
173	lastNotifyRecord
174	scheduleDefault
175	acceptedModes
176	adjustValue
177	count
178	countBeforeChange
179	countChangeTime
180	covPeriod
181	inputReference
182	limitMonitoringInterval

183	loggingObject
184	loggingRecord
185	prescale
186	pulseRate
187	scale
188	scaleFactor
189	updateTime
190	valueBeforeChange
191	valueSet
192	valueChangeTime
193	alignIntervals
195	intervalOffset
196	lastRestartReason
197	loggingType
202	restartNotificationRecipients
203	timeOfDeviceRestart
204	timeSynchronizationInterval
205	trigger
206	utcTimeSynchronizationRecipients
207	nodeSubtype
208	nodeType
209	structuredObjectList
210	subordinateAnnotations
211	subordinateList
212	actualShedLevel
213	dutyWindow
214	expectedShedLevel
215	fullDutyBaseline
218	requestedShedLevel
219	shedDuration
220	shedLevelDescriptions
221	shedLevels
222	stateDescription
226	doorAlarmState
227	doorExtendedPulseTime
228	doorMembers
229	doorOpenTooLongTime
230	doorPulseTime
231	doorStatus

232	doorUnlockDelayTime
233	lockStatus
234	maskedAlarmValues
235	securedStatus
244	absenteeLimit
245	accessAlarmEvents
246	accessDoors
247	accessEvent
248	accessEventAuthenticationFactor
249	accessEventCredential
250	accessEventTime
251	accessTransactionEvents
252	accompaniment
253	accompanimentTime
254	activationTime
255	activeAuthenticationPolicy
256	assignedAccessRights
257	authenticationFactors
258	authenticationPolicyList
259	authenticationPolicyNames
260	authenticationStatus
261	authorizationMode
262	belongsTo
263	credentialDisable
264	credentialStatus
265	credentials
266	credentialsInZone
267	daysRemaining
268	entryPoints
269	exitPoints
270	expirationTime
271	extendedTimeEnable
272	failedAttemptEvents
273	failedAttempts
274	failedAttemptsTime
275	lastAccessEvent
276	lastAccessPoint
277	lastCredentialAdded
278	lastCredentialAddedTime

279	lastCredentialRemoved
280	lastCredentialRemovedTime
281	lastUseTime
282	lockout
283	lockoutRelinquishTime
285	maxFailedAttempts
286	members
287	musterPoint
288	negativeAccessRules
289	numberOfAuthenticationPolicies
290	occupancyCount
291	occupancyCountAdjust
292	occupancyCountEnable
294	occupancyLowerLimit
295	occupancyLowerLimitEnforced
296	occupancyState
297	occupancyUpperLimit
298	occupancyUpperLimitEnforced
300	passbackMode
301	passbackTimeout
302	positiveAccessRules
303	reasonForDisable
304	supportedFormats
305	supportedFormatClasses
306	threatAuthority
307	threatLevel
308	traceFlag
309	transactionNotificationClass
310	userExternalIdentifier
311	userInformationReference
317	userName
318	userType
319	usesRemaining
320	zoneFrom
321	zoneTo
322	accessEventTag
323	globalIdentifier
326	verificationTime
327	baseDeviceSecurityPolicy

328	distributionKeyRevision
329	doNotHide
330	keySets
331	lastKeyServer
332	networkAccessSecurityPolicies
333	packetReorderTime
334	securityPduTimeout
335	securityTimeWindow
336	supportedSecurityAlgorithms
337	updateKeySetTimeout
338	backupAndRestoreState
339	backupPreparationTime
340	restoreCompletionTime
341	restorePreparationTime
342	bitMask
343	bitText
344	isUtc
345	groupMembers
346	groupMemberNames
347	memberStatusFlags
348	requestedUpdateInterval
349	covuPeriod
350	covuRecipients
351	eventMessageTexts
352	eventMessageTextsConfig
353	eventDetectionEnable
354	eventAlgorithmInhibit
355	eventAlgorithmInhibitRef
356	timeDelayNormal
357	reliabilityEvaluationInhibit
358	faultParameters
359	faultType
360	localForwardingOnly
361	processIdentifierFilter
362	subscribedRecipients
363	portFilter
364	authorizationExemptions
365	allowGroupDelayInhibit
366	channelNumber

367 controlGroups
368 executionDelay
369 lastPriority
370 writeStatus
371 propertyList
372 serialNumber
373 blinkWarnEnable
374 defaultFadeTime
375 defaultRampRate
376 defaultStepIncrement
377 egressTime
378 inProgress
379 instantaneousPower
380 lightingCommand
381 lightingCommandDefaultPriority
382 maxActualValue
383 minActualValue
384 power
385 transition
386 egressActive
387 interfaceValue
388 faultHighLimit
389 faultLowLimit
390 lowDiffLimit
391 strikeCount
392 timeOfStrikeCountReset
393 defaultTimeout
394 initialTimeout
395 lastStateChange
396 stateChangeValues
397 timerRunning
398 timerState
399 apduLength
400 ipAddress
401 ipDefaultGateway
402 ipDhcpEnable
403 ipDhcpLeaseTime
404 ipDhcpLeaseTimeRemaining
405 ipDhcpServer

406	ipDnsServer
407	bacnetIpGlobalAddress
408	bacnetIpMode
409	bacnetIpMulticastAddress
410	bacnetIpNatTraversal
411	ipSubnetMask
412	bacnetIpUdpPort
413	bbmdAcceptFdRegistrations
414	bbmdBroadcastDistributionTable
415	bbmdForeignDeviceTable
416	changesPending
417	command
418	fdBbmdAddress
419	fdSubscriptionLifetime
420	linkSpeed
421	linkSpeeds
422	linkSpeedAutonegotiate
423	macAddress
424	networkInterfaceName
425	networkNumber
426	networkNumberQuality
427	networkType
428	routingTable
429	virtualMacAddressTable
430	commandTimeArray
431	currentCommandPriority
432	lastCommandTime
433	valueSource
434	valueSourceArray
435	bacnetIpv6Mode
436	ipv6Address
437	ipv6PrefixLength
438	bacnetIpv6UdpPort
439	ipv6DefaultGateway
440	bacnetIpv6MulticastAddress
441	ipv6DnsServer
442	ipv6AutoAddressingEnable
443	ipv6DhcpLeaseTime
444	ipv6DhcpLeaseTimeRemaining

445	ipv6DhcpServer
446	ipv6ZoneIndex
447	assignedLandingCalls
448	carAssignedDirection
449	carDoorCommand
450	carDoorStatus
451	carDoorText
452	carDoorZone
453	carDriveStatus
454	carLoad
455	carLoadUnits
456	carMode
457	carMovingDirection
458	carPosition
459	elevatorGroup
460	energyMeter
461	energyMeterRef
462	escalatorMode
463	faultSignals
464	floorText
465	groupId
467	groupMode
468	higherDeck
469	installationId
470	landingCalls
471	landingCallControl
472	landingDoorStatus
473	lowerDeck
474	machineRoomId
475	makingCarCall
476	nextStoppingFloor
477	operationDirection
478	passengerAlarm
479	powerMode
480	registeredCarCall
481	activeCovMultipleSubscriptions
482	protocolLevel
483	referencePort
484	deployedProfileLocation

485	profileLocation
486	tags
487	subordinateNodeTypes
488	subordinateTags
489	subordinateRelationships
490	defaultSubordinateRelationship
491	represents

BACnetProtocolLevel

This enumeration is used by the following objects/properties:

- **networkPort.protocolLevel**

The enumeration values are:

Value	Name
0	physical
1	protocol
2	bacnetApplication
3	nonBacnetApplication

BACnetRelationship

This enumeration is used by the following objects/properties:

- **structuredView.defaultSubordinateRelationship**

The enumeration values are:

Value	Name
0	unknown
1	default
2	contains
3	containedBy
4	uses
5	usedBy
6	commands
7	commandedBy
8	adjusts
9	adjustedBy
10	ingress
11	egress
12	suppliesAir
13	receivesAir
14	suppliesHotAir
15	receivesHotAir
16	suppliesCoolAir
17	receivesCoolAir
18	suppliesPower
19	receivesPower
20	suppliesGas
21	receivesGas
22	suppliesWater
23	receivesWater
24	suppliesHotWater
25	receivesHotWater
26	suppliesCoolWater
27	receivesCoolWater
28	suppliesSteam
29	receivesSteam

BACnetReliability

This enumeration is used by the following objects/properties:

- **accessCredential.reliability**
- **accessDoor.reliability**
- **accessPoint.reliability**
- **accessRights.reliability**

- **accessUser.reliability**
- **accessZone.reliability**
- **accumulator.reliability**
- **analogInput.reliability**
- **analogOutput.reliability**
- **analogValue.reliability**
- **binaryInput.reliability**
- **binaryLightingOutput.reliability**
- **binaryOutput.reliability**
- **binaryValue.reliability**
- **bitstringValue.reliability**
- **channel.reliability**
- **characterstringValue.reliability**
- **command.reliability**
- **credentialDataInput.reliability**
- **dateValue.reliability**
- **datepatternValue.reliability**
- **datetimeValue.reliability**
- **datetimepatternValue.reliability**
- **device.reliability**
- **escalator.reliability**
- **eventEnrollment.reliability**
- **eventLog.reliability**
- **globalGroup.reliability**
- **integerValue.reliability**
- **largeAnalogValue.reliability**
- **lifeSafetyPoint.reliability**
- **lifeSafetyZone.reliability**
- **lift.reliability**
- **lightingOutput.reliability**
- **loadControl.reliability**
- **loop.reliability**

- **multiStateInput.reliability**
- **multiStateOutput.reliability**
- **multiStateValue.reliability**
- **networkPort.reliability**
- **notificationClass.reliability**
- **notificationForwarder.reliability**
- **octetstringValue.reliability**
- **positiveIntegerValue.reliability**
- **program.reliability**
- **pulseConverter.reliability**
- **schedule.reliability**
- **timeValue.reliability**
- **timepatternValue.reliability**
- **timer.reliability**
- **trendLog.reliability**
- **trendLogMultiple.reliability**

The enumeration values are:

Value	Name
0	noFaultDetected
1	noSensor
2	overRange
3	underRange
4	openLoop
5	shortedLoop
6	noOutput
7	unreliableOther
8	processError
9	multiStateFault
10	configurationError
12	communicationFailure
13	memberFault
14	monitoredObjectFault
15	tripped
16	lampFailure
17	activationFailure
18	renewDhcpFailure
19	renewFdRegistrationFailure
20	restartAutoNegotiationFailure
21	restartFailure
22	proprietaryCommandFailure
23	faultsListed
24	referencedObjectFault

BACnetRestartReason

This enumeration is used by the following objects/properties:

- **device.lastRestartReason**

The enumeration values are:

Value	Name
0	unknown
1	coldstart
2	warmstart
3	detectedPowerLost
4	detectedPoweredOff
5	hardwareWatchdog
6	softwareWatchdog
7	suspended
8	activateChanges

BACnetSecurityLevel

This enumeration is used by the following objects/properties:

- **networkSecurity.baseDeviceSecurityPolicy**

The enumeration values are:

Value	Name
0	incapable
1	plain
2	signed
3	encrypted
4	signedEndToEnd
5	encryptedEndToEnd

BACnetSegmentation

This enumeration is used by the following objects/properties:

- **device.segmentationSupported**

The enumeration values are:

Value	Name
0	segmentedBoth
1	segmentedTransmit
2	segmentedReceive
3	noSegmentation

BACnetShedState

This enumeration is used by the following objects/properties:

- **loadControl.presentValue**

The enumeration values are:

Value	Name
0	shedInactive
1	shedRequestPending
2	shedCompliant
3	shedNonCompliant

BACnetSilencedState

This enumeration is used by the following objects/properties:

- **lifeSafetyPoint.silenced**
- **lifeSafetyZone.silenced**

The enumeration values are:

Value	Name
0	unsilenced
1	audibleSilenced
2	visibleSilenced
3	allSilenced

BACnetTimerState

This enumeration is used by the following objects/properties:

- **timer.timerState**

The enumeration values are:

Value	Name
0	idle
1	running
2	expired

BACnetTimerTransition

This enumeration is used by the following objects/properties:

- **timer.lastStateChange**

The enumeration values are:

Value	Name
0	none
1	idleToRunning
2	runningToIdle
3	runningToRunning
4	runningToExpired
5	forcedToExpired
6	expiredToIdle
7	expiredToRunning

BACnetWriteStatus

This enumeration is used by the following objects/properties:

- **channel.writeStatus**

The enumeration values are:

Value	Name
0	idle
1	inProgress
2	successful
3	failed

Driver Revision History

VERSION	DATE	AUTHOR	COMMENTS
1.2.117	12/12/2024	M. Ludwig A. Fetzner	• Driver updated to the most recent version of IOKit (<i>Case 37039</i>). • Fixed a failure when listing objects and properties in the Tag Browser for devices that do not support packet segmentation (<i>Case 36482</i>).
1.2.116	07/10/2024	M. Ludwig	• Documentation updates.
1.2.112	05/02/2024	M. Ludwig A. Fetzner	• Implemented the rejection of specific messages only for BBMD devices (<i>Case 35540</i>). • Implemented sending and receiving segmented messages, allowing reading and writing long properties (<i>Case 29389</i>).
1.2.111	04/18/2024	M. Ludwig A. Fetzner	• Added information about sources in Remote

VERSION	DATE	AUTHOR	COMMENTS
			Broadcast-type messages (Case 35824). - Implemented the rejection of specific messages only for router devices (Case 35794). - Fixed the increment of <i>Invoke ID</i> in each retry (Case 29137).
1.2.96	06/16/2021	G. Beal	- Improved the Alarm and Event functionality, by adding the GetEventInformation and GetAlarmSummary services, and enhancing the GetEnrollmentSummary and AcknowledgeAlarm services (Case 29049). - Implemented properties for this Driver's <i>Device</i> object, with configurable DeviceName on the properties page (Case 29059).
1.2.95	05/18/2021	G. Beal	- Implemented a way of sending the WritePropertyMultiple service using special Tags (Case 29048). - Implemented a way of sending the ReadRange service using special Tags (Case 29050). - Fixed the writing of a Null value in commanding properties (Case 30925).
1.2.94	02/23/2021	G. Beal	- Added the Auto-Detect Devices Range configuration (Case 29046). - Implemented a way of sending TimeSynchronization and UTCTimeSynchronization services using special Tags (Case 29053). - In Tag Browser, inside the Network Browser node, after expanding an object node, this Driver now provides only properties that actually exist within an object, obtained by reading

VERSION	DATE	AUTHOR	COMMENTS
			its propertyList property (Case 30082). Added a new folder in the Tag Browser where the devices found by the Driver are configured in Elipse BMS mode (Case 29294).
1.2.89	11/30/2020	G. Beal	- Dates and times containing special values can now be displayed and written using the XML format (Case 29895). - Added the Read Quality-related Properties configuration (Case 29955).
1.2.87	10/22/2020	F. Englert G. Beal	- Fixed this Driver's response to reading requests of properties with a BACnetARRAY data type with an index (Case 29431). - Added the delay configurations Inter-Message Delay, SubscribeCOV Delay, and Network Delay (Case 29574).
1.2.86	08/28/2020	F. Englert G. Beal	- Updated the value of the vendorIdentifier property in this Driver's <i>Device</i> object to the value 1245, assigned to Elipse Software by ASHRAE (Case 28930). - This Driver now responds to received Who-Has messages if the requested object is available (Case 29052). - Fixed the handling of received Who-Is messages, so this Driver now responds only if the identifier is within the requested range (Case 29055). - Fixed the handling of confirmed requests received with encoding errors, so this Driver now responds with a Reject-PDU error (Case 29151). - Fixed the handling of reading requests for a Device-4194303 object,

VERSION	DATE	AUTHOR	COMMENTS
			wildcard value, so this Driver now responds using its proper device identifier (Case 29154). - Fixed this Driver's response to reading requests of non-existing objects, now responding with an error class OBJECT and an error code UNKNOWN_OBJECT (Case 29155). - Fixed this Driver's response to reading requests with index to properties that do not have a BACnetARRAY data type, now responding with an error class PROPERTY and an error code PROPERTY_IS_NOT_AN_ARRAY (Case 29158). - Fixed the initialization of NPDU's HopCount field to the value 255 on packets sent to sub-networks (Case 29342). - This Driver now ignores packets received in situations of confirmed requests sent in broadcast and in requests sent to a network address different from this Driver's address (Case 29343). - This Driver now correctly responds to reading requests of all, optional and required special properties (Case 29375).
1.2.82	01/28/2020	F. Englert	This Driver now responds to the ReadPropertyMultiple service (Case 28173).
1.2.79	12/05/2019	F. Englert	This Driver now accepts readings requested by the Device-4194303 object, handling this value as a wildcard to access its <i>Device</i> object (Case 28086).
1.2.76	06/11/2019	F. Englert	This Driver's Device Instance can now be configured on the configuration window (Case 26902).

VERSION	DATE	AUTHOR	COMMENTS
1.2.68	4/25/2019	F. Englert	This Driver now supports reading and writing any property. Properties with complex structures use an XML format (<i>Case 23292</i>).
1.2.53	05/04/2017	F. Englert	Added the Exclude Analog objects from COV configuration (<i>Case 22639</i>).
1.2.51	03/20/2017	F. Englert	This Driver's COV Process Identifier can now be configured on the configuration window (<i>Case 22366</i>).
1.2.49	12/01/2016	F. Englert	Fixed the introductory text of this Driver's documentation (<i>Case 11494</i>).
1.2.43	12/03/2015	F. Englert	Added the Maximum Transactions per Network configuration on this Driver to limit the maximum number of simultaneous communications with each network (<i>Case 20282</i>).
1.2.41	08/26/2015	F. Englert	Added the BACnet.Retries property, which allows defining the number of retries for requests that fail by time-out (<i>Case 18429</i>).
1.2.40	07/24/2015	F. Englert	- This Driver's configuration window was redesigned (<i>Case 19303</i>). - Runtime Browse now works in devices without support for packet segmentation (<i>Case 19151</i>).
1.2.38	07/13/2015	F. Englert	- This Driver now changes the quality of Tags that read the presentValue property according to the current value of statusFlags and reliability properties (<i>Case 18704</i>). - Fixed the behavior of Change Of Value (COV) subscriptions, so that this Driver cancels subscriptions of Tags not being read by an application anymore (<i>Case 19164</i>).

VERSION	DATE	AUTHOR	COMMENTS
1.2.36	06/26/2015	F. Englert	The Tag Browser tool now generates Tags with the name of the object configured in the device (<i>Case 18738</i>).
1.2.35	06/18/2015	F. Englert	Reading a list of objects from a device now works in steps, allowing to execute a browse of devices that do not support packet segmentation (<i>Case 18874</i>).
1.2.34	05/04/2015	F. Englert	This Driver now supports device addresses greater than 65535 (<i>Case 18513</i>).
1.2.33	04/27/2015	F. Englert	This Driver's auto-discovery procedure now works correctly with any IP address, not only with the address 255.255.255.255, and does not require a BACnet router anymore (<i>Case 18010</i>).
1.2.32	04/27/2015	F. Englert	This Driver now allows communicating with several IP addresses at the same time, by configuring the Ethernet interface to the listen mode (<i>Case 18012</i>).
1.2.31	03/10/2014	F. Englert	The optional padding byte at the end of BACnet MS/TP protocol packets now is handled as truly optional by this Driver, which allows communicating without problems with devices that do not send this byte (<i>Case 15853</i>).
1.2.29	09/13/2013	F. Englert	Added support for the Schedule.scheduleDefault property (<i>Case 14885</i>).
1.2.28	07/04/2013	F. Englert	- Fixed a problem on readings and writings of the schedule.exceptionSchedule property, which may or may not work depending on the data to read or write (<i>Case 13780</i>). - Fixed a problem on writings of the exceptionSchedule property, which failed if one of the elements was a date (<i>Case 13911</i>).

VERSION	DATE	AUTHOR	COMMENTS
1.2.27	03/25/2013	F. Englert	Fixed the reading of the Calendar.DateList property, which could display incomplete values (Case 13907).
1.2.26	03/20/2013	F. Englert	- Driver ported to IOKit 2.0 (Case 13853). - The value configured in the Max Items per ReadPropertyMultiple Request field on a device's editing window now works correctly. It always used the default value 50 (Case 13885).
1.2.25	02/20/2013	F. Englert	Fixed a memory leak that occurred when this Driver sent or received segmented messages whose aggregated size exceeded 1024 bytes (Case 13741).
1.2.24	02/06/2013	M. Salvador	- Implemented support for COV and Alarm&Events (Case 13537). - Fixed a problem in the implementation of the Multistate.presentValue property, which did not allow cleaning the property with an Empty value (Case 13690).
1.2.13	04/20/2011	F. Englert	- This Driver now supports grouping several readings on the same command, considerably improving performance (Case 11596). - Added support for BACnet MS/TP protocol (Case 12004).
1.2.10	06/22/2010	F. Englert	Added support for reading and writing complex properties, such as schedule.weeklySchedule (Case 10876).
1.2.5	04/19/2010	C. Mello	Generated a test version of this Driver for Windows CE platform (Case 10797).
1.2.4	08/17/2009	F. Englert	Now this Driver allows writing a NULL value to the presentValue property of analogOutput,

VERSION	DATE	AUTHOR	COMMENTS
			analogValue, binaryOutput, binaryValue, and multistateOutput objects (Case 10722). - Fixed a Driver's failure after receiving an unsolicited or duplicated message (Case 10071). - This Driver now supports accessing objects with an address above 65535 in a configuration by Strings (Case 10543).
1.2.3	08/07/2009	F. Englert	Added support for reading in the priorityArray property (Case 10689).
1.1.1	09/10/2007	F. Englert	- Fixed an error when writing values (Case 7076). - Fixed the validation of values in properties with enumerations (Case 7077). - Fixed a misspelling error on BACnet tab (Case 7091). - Added an option to indicate a priority for writings of this Driver (Case 8425).
1.0.1	02/06/2006	F. Englert	Driver's initial version (Case 5437).

Headquarters

**Rua Mostardeiro, 322/Cj. 902, 1001 e
1002**

90510-002 — Porto Alegre — RS

Phone: (+55 51) 3346-4699

Fax: (+55 51) 3222-6226

E-mail: elipse-rs@elipse.com.br

Branch in Taiwan

9F., No.12, Beiping 2nd St., Sanmin Dist.

807 — Kaohsiung City — Taiwan

Phone: (+886 7) 323-8468

Fax: (+886 7) 323-9656

E-mail: evan@elipse.com.br

Check our website for information about a representative in your country.

www.elipse.com.br

kb.elipse.com.br

forum.elipse.com.br

www.youtube.com/elipsesoftware

elipse@elipse.com.br



Gartner, Cool Vendors in Brazil 2014, April 2014.

Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability of fitness for a particular purpose.

Microsoft Partner

Gold Independent Software Vendor (ISV)