



---

**INDUSTRIAL CONTROL COMMUNICATIONS, INC.**

---

# **BACnet MS/TP Server Driver Manual**

---



# TABLE OF CONTENTS

- 1 BACnet MS/TP Server..... 2**
  - 1.1 Overview ..... 2
  - 1.2 Supported BACnet Services ..... 2
  - 1.3 Server Settings..... 4
  - 1.4 Node Settings..... 4
  - 1.5 Device Object Settings..... 5
    - 1.5.1 OEM Settings (PicoPort/Mirius Only) ..... 5
    - 1.5.2 Real-time Clock Settings (PicoPort/Mirius Only) ..... 5
  - 1.6 BACnet Object Settings ..... 7
    - 1.6.1 Default BACnet Mapping..... 7
      - 1.6.1.1 Default BACnet Analog Objects..... 8
      - 1.6.1.2 Default BACnet Binary Objects..... 9
      - 1.6.1.3 Default BACnet Multi-state Objects ..... 10
    - 1.6.2 Analog Input Object Settings ..... 12
    - 1.6.3 Analog Output Object Settings ..... 13
    - 1.6.4 Analog Value Object Settings..... 14
    - 1.6.5 Binary Input Object Settings..... 15
    - 1.6.6 Binary Output Object Settings ..... 15
    - 1.6.7 Binary Value Object Settings..... 17
    - 1.6.8 Multi-state Input Object Settings..... 18
    - 1.6.9 Multi-state Output Object Settings..... 18
    - 1.6.10 Multi-state Value Object Settings ..... 19
  - 1.7 Relinquish Default Event (PicoPort/Mirius Only)..... 20

# 1 BACnet MS/TP Server

## 1.1 Overview

This driver supports the BACnet MS/TP server protocol. The MS/TP server driver acts as a master on the network, meaning it is actively involved in token management.

Some notes of interest are:

- The ability to add predefined default BACnet objects provides flexible and convenient access to the entire internal database.
- The APDU Timeout, Number of APDU Retries, and Max Master properties of the Device object are writeable over the BACnet network.
- The Device Name, Device Instance, and Max Info Frames properties of the Device object are writeable over the BACnet network (PicoPort/Mirius only).
- The COV Increment property of Analog Input, Analog Output, and Analog Value objects is writeable over the BACnet network for devices that support COV reporting.
- Fully configurable and discoverable user-defined BACnet objects.
- Supported BACnet objects include:
  - Device
  - Analog input (AI), output (AO) and value (AV)
  - Binary input (BI), output (BO) and value (BV)
  - Multi-state input (MSI), output (MSO) and value (MSV)
- Supports MS/TP baud rates from 9600 baud to 115200 baud.
- Binary Objects support custom Active and Inactive Text.
- Whenever a node is added to the project, its BACnet device object is always present and must be properly configured.

## 1.2 Supported BACnet Services

This section details the BACnet services that are supported by the driver.

### **Read Property**

This service is used to request data from the device pertaining to a single BACnet object property.

### **Read Property Multiple**

This service is used to request data from the device pertaining to several BACnet object properties.

### **Write Property**

This service is used to modify the value of a single BACnet object property.



### **Write Property Multiple**

This service is used to modify the values of several BACnet object properties.

### **COV Reporting**

This service is used to provide change of value notifications to clients that subscribe to them. This is an alternative option to using the Read Property or Read Property Multiple services to monitor data on the device. Note that this service is not supported by all devices. Please refer to the device's BACnet Protocol Implementation Conformance Statement (PICS) for information on whether this service is supported.

### **Dynamic Device Binding**

This service is used to discover the device on the network. Upon receiving a "Who-Is" request on the network, the device will generate an "I-Am" response, provided its assigned instance number is included within the range of the request. This allows clients to resolve the device's network address.

### **Dynamic Object Binding**

This service is used to discover the device's objects on the network. Upon receiving a "Who-Has" request on the network, the device will generate an "I-Have" response, provided the targeted object exists on the device. This allows clients to resolve which devices on the network have specific BACnet objects.

### **Device Communication Control**

This service is used to halt responses to requests directed at the device for a defined amount of time (or indefinitely). Once communication is disabled, the device will only respond to a device communication control service that re-enables communication or a reinitialize device service that resets the device until the defined amount of time has expired. This service is password protected. To successfully enable or disable communications on the device, "icc" must be used as the password. This service is generally only used for commissioning purposes.

### **Reinitialize Device**

This service is used to reset the device. The driver does not distinguish between a warm and cold restart. This service is password protected. To successfully reset the device, "icc" must be used as the password.

### **Time Synchronization**

This service is used to notify the device of the correct current time and date. Note that this service is only supported when Real-time Clock Settings are enabled and is not supported by all devices. Please refer to the device's BACnet Protocol Implementation Conformance Statement (PICS) for information on whether this service is supported.

### **UTC Time Synchronization**

This service is used to notify the device of the correct Coordinated Universal Time (UTC). Note that this service is only supported when Real-time Clock Settings are enabled and is not supported by all devices. Please refer to the device's BACnet Protocol Implementation Conformance Statement (PICS) for information on whether this service is supported.



## 1.3 Server Settings

### **Baud Rate**

Selects the baud rate of the network.

### **Parity**

Fixed at No Parity (1 Stop Bit).

### **APDU Timeout**

Sets the time in milliseconds that the driver will wait for a response from a device after sending a request. As a BACnet server, this setting only applies to Confirmed COV Notifications. Note that this setting is only available for devices that support COV reporting.

### **Number of APDU Retries**

Sets the number of times that the driver will retry a request when a response is not received. As a BACnet server, this setting only applies to Confirmed COV Notifications. Note that this setting is only available for devices that support COV reporting.

### **Max Master**

Defines the highest allowable address (0...127) for MS/TP master nodes on the network. Any address higher than this will not receive the token from the driver. Note that this value must be greater than or equal to the driver's configured MAC Address(es). If the highest address on the network is unknown, set this field to 127.

### **Max Info Frames (PicoPort/Mirius Only)**

Defines the maximum number of information frames the device may send before it must pass the token.

### **Fail-safe Timeout Time (PicoPort/Mirius Only)**

Defines the maximum number of milliseconds for a break in network communications before a timeout event will be triggered. To disable timeout processing, set this field to 0.

Note that due to the nature of BACnet MS/TP, there is always token passing traffic which does not necessarily indicate the presence of a client. Therefore, "network communications" in the context of timeout processing is defined as the device receiving BACnet Confirmed Request PDUs from the client.

## 1.4 Node Settings

### **MAC Address**

Defines the address (0...127) that the node will reside at on the network.



## 1.5 Device Object Settings

A Device Object is automatically added to every node, and cannot be removed. The Device Object contains several configurable fields that must be appropriately set for each device residing on a BACnet network.

### **Device Name**

Defines the node's name. The device name must be unique across the entire BACnet network. Enter a string of between 1 and 16 characters in length.

### **Instance Number**

Defines the node's instance number. The instance number must be unique across the entire BACnet network. Enter a value between 0...4194302 (0x0...0x3FFFFE).

### 1.5.1 OEM Settings (PicoPort/Mirius Only)

The following fields allow an OEM to configure the device to appear as their product on the BACnet network.

#### **Vendor ID**

Specifies the BACnet Vendor ID assigned to the OEM. The default value is 242, ICC's BACnet Vendor ID.

#### **Vendor Name**

Specifies the name of the OEM. The default value is "ICC, Inc."

#### **Model Name**

Specifies the model name of the OEM's product. The default value is the ICC product name.

### 1.5.2 Real-time Clock Settings (PicoPort/Mirius Only)

The real-time clock settings allow the device to utilize time and date information residing in the device's database. When enabled, the device will expose the Local Time, Local Date, UTC Offset, and Daylight Savings Status device object properties as well as support the Time Synchronization and UTC Time Synchronization services.

Note that the device itself does not have a real-time clock. It is expected that a host device populates the device's database with real-time clock data.

#### **Enable RTC**

Check this option to enable the real-time clock settings.

#### **RTC Format**

Select the real-time clock data format used to represent time and date information in the device's database. The available options are detailed below.

##### **Date-Time Format**

This option uses eight bytes to represent the date and time information. The meaning of each byte is described in Table 1.



**Table 1: Date-Time Format**

Byte Offset	Value	Description
0	Year	The year minus 1900
1	Month	The month (1...12, where January = 1)
2	Day	The day of the month (1...31)
3	Weekday	The day of the week (1...7, where Monday = 1)
4	Hour	The hour in the 24-hour system (0...23)
5	Minute	The minute of the hour (0...59)
6	Second	The second of the minute (0...59)
7	Hundredths	The fractional part of the second in hundredths of a second (0...99)

**Unix Time Format**

This option uses a single 32-bit value to represent both date and time information. The value is defined as the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, January 1, 1970, not counting leap seconds.

**RTC Database Address**

Defines the starting address in the database where the date and time information is mapped.

**Time Zone**

Select the time zone for the location in which the device is installed.

**Daylight Saving Time**

Check this option to indicate whether daylight saving time is currently in effect at the location in which the device is installed.

**1.6 BACnet Object Settings**

The BACnet server hosts BACnet objects which contain many different properties for any BACnet client on the network to access. The driver supports a variety of different BACnet objects (refer to section 1.1). All supported properties of these objects are readable, while only the present value property is writable (for Outputs and Values only).

**1.6.1 Default BACnet Mapping**

When the “Default BACnet Mapping” item is added to the node, predefined BACnet objects are automatically mapped to the internal database to provide convenient bit, byte (8-bit signed), word (16-bit signed), and double word (32-bit signed) accessibility. Note that these default objects are not discoverable using the “Who-Has” service, or via the object list property of the device object. The properties of these default objects can only be accessed using the Read Property, Read Property Multiple, Write Property, and Write Property Multiple services. It is expected that only the Present Value property will be accessed, as the other properties do not

provide any useful information. The default objects are not configurable and do not support COV, the Priority Array, or optional properties. User-defined BACnet objects can co-exist with the default BACnet objects.

### 1.6.1.1 Default BACnet Analog Objects

Analog Input, Analog Output, and Analog Value objects are automatically mapped to the database. The Analog object's instance number will determine the data type access and the database address. Refer to Table 2.

**Table 2: Default BACnet Analog Object Mapping**

Instance Number	Data Type	Database Address Equation
0...(DB <sub>Size</sub> - 1)	Byte (8-bit signed)	Equation 1
DB <sub>Size</sub> ... (2 × DB <sub>Size</sub> ) - 2	Word (16-bit signed)	Equation 2
(2 × DB <sub>Size</sub> ) ... (3 × DB <sub>Size</sub> ) - 4	Double Word (32-bit signed)	Equation 3

Where DB<sub>Size</sub> is the size of the database on the device.

$$\text{Database Address} = \text{Instance Number} \quad \text{Equation 1}$$

$$\text{Database Address} = \text{Instance Number} - \text{DB}_{\text{Size}} \quad \text{Equation 2}$$

$$\text{Database Address} = \text{Instance Number} - (2 \times \text{DB}_{\text{Size}}) \quad \text{Equation 3}$$

The Analog object names are generated according to the following structure:

**[object type]\_[data type]\_[address]**

Where

**[object type]** is a 2-character field: "AI" for Analog Input, "AO" for Analog Output, and "AV" for Analog Value.

**[data type]** is a 1-character field: "b" for byte access, "w" for word access, and "d" for double word access.

**[address]** is a 1- to 4-character field ("0", "1", "2", etc.) corresponding to the referenced starting database address.

**Table 3: Default BACnet Analog Object Byte Access Examples**

Description	Object Type	Instance	Name
Read one byte starting at database address 0	Analog Input	0	AI_b_0
	Analog Output		AO_b_0
	Analog Value		AV_b_0
Write one byte starting at	Analog Output	1000	AO_b_1000

Description	Object Type	Instance	Name
database address 1000	Analog Value		AV_b_1000

**Table 4: Default BACnet Analog Object Word Access Examples**

Description	Object Type	Instance	Name
Read one word starting at database address 100	Analog Input	4196	AI_w_100
	Analog Output		AO_w_100
	Analog Value		AV_w_100
Write one word starting at database address 2002	Analog Output	6098	AO_w_2002
	Analog Value		AV_w_2002

Note that this example assumes  $DB_{Size} = 4096$

**Table 5: Default BACnet Analog Object Double Word Access Examples**

Description	Object Type	Instance	Name
Read one double word starting at database address 4	Analog Input	8196	AI_d_4
	Analog Output		AO_d_4
	Analog Value		AV_d_4
Write one double word starting at database address 4092	Analog Output	12284	AO_d_4092
	Analog Value		AV_d_4092

Note that this example assumes  $DB_{Size} = 4096$

### 1.6.1.2 Default BACnet Binary Objects

Binary Input, Binary Output, and Binary Value objects are automatically mapped to individual bits in the database. The Binary object instance numbers range from  $0 \dots (8 \times DB_{Size}) - 1$ , where  $DB_{Size}$  is the size of the database on the device, and will determine the database address and the corresponding bit. To calculate the database address and the bit offset, refer to Equation 4 and Equation 5, respectively.

$$Database\ Address = (Instance\ Number) / 8 \quad \text{Equation 4}$$

$$Bit\ Offset = (Instance\ Number) \% 8 \quad \text{Equation 5}$$

The Binary object names are generated according to the following structure:

**[object type]\_[address]\_[bit]**

*Where*

**[object type]** is a 2-character field: “BI” for Binary Input, “BO” for Binary Output, and “BV” for Binary Value.

[address] is a 1- to 4-character field (“0”, “1”, “2”, etc.) corresponding to the referenced database address.

[bit] is a 1-character field (“0”, “1”...“6”, “7”) corresponding to the bit offset.

**Table 6: Default BACnet Binary Object Access Examples**

Description	Object Type	Instance	Name
Read database address 0, bit 0	Binary Input	0	BI_0_0
	Binary Output		BO_0_0
	Binary Value		BV_0_0
Read database address 0, bit 7	Binary Input	7	BI_0_7
	Binary Output		BO_0_7
	Binary Value		BV_0_7
Read database address 4095, bit 6	Binary Input	32766	BI_4095_6
	Binary Output		BO_4095_6
	Binary Value		BV_4095_6
Write database address 1, bit 1	Binary Output	9	BO_1_1
	Binary Value		BV_1_1
Write database address 100, bit 3	Binary Output	803	BO_803_3
	Binary Value		BV_803_3
Write database address 4095, bit 7	Binary Output	32767	BO_4095_7
	Binary Value		BV_4095_7

### 1.6.1.3 Default BACnet Multi-state Objects

Multi-state Input, Multi-state Output, and Multi-state Value objects are automatically mapped to the database. The Multi-state object’s instance number will determine the data type access and the database address. Refer to Table 7.

**Table 7: Default BACnet Multi-state Object Mapping**

Instance Number	Data Type	Database Address Equation
0...(DB <sub>Size</sub> - 1)	Byte (8-bit signed)	Equation 6
DB <sub>Size</sub> ... (2 × DB <sub>Size</sub> ) - 2	Word (16-bit signed)	Equation 7
(2 × DB <sub>Size</sub> ) ... (3 × DB <sub>Size</sub> ) - 4	Double Word (32-bit signed)	Equation 8

Where DB<sub>Size</sub> is the size of the database on the device.

$$\text{Database Address} = \text{Instance Number} \quad \text{Equation 6}$$

$$\text{Database Address} = \text{Instance Number} - \text{DB}_{\text{Size}} \quad \text{Equation 7}$$

$$\text{Database Address} = \text{Instance Number} - (2 \times \text{DB}_{\text{Size}})$$

**Equation 8**

The Multi-state object names are generated according to the following structure:

**[object type]\_[data type]\_[address]**

Where

**[object type]** is a 3-character field: “MSI” for Multi-state Input, “MSO” for Multi-state Output, and “MSV” for Multi-state Value.

**[data type]** is a 1-character field: “b” for byte access, “w” for word access, and “d” for double word access.

**[address]** is a 1- to 4-character field (“0”, “1”, “2”, etc.) corresponding to the referenced starting database address.

**Table 8: Default BACnet Multi-state Object Byte Access Examples**

Description	Object Type	Instance	Name
Read one byte starting at database address 0	Multi-state Input	0	MSI_b_0
	Multi-state Output		MSO_b_0
	Multi-state Value		MSV_b_0
Write one byte starting at database address 1000	Multi-state Output	1000	MSO_b_1000
	Multi-state Value		MSV_b_1000

**Table 9: Default BACnet Multi-state Object Word Access Examples**

Description	Object Type	Instance	Name
Read one word starting at database address 100	Multi-state Input	4196	MSI_w_100
	Multi-state Output		MSO_w_100
	Multi-state Value		MSV_w_100
Write one word starting at database address 2002	Multi-state Output	6098	MSO_w_2002
	Multi-state Value		MSV_w_2002

Note that this example assumes  $\text{DB}_{\text{Size}} = 4096$

**Table 10: Default BACnet Multi-state Object Double Word Access Examples**

Description	Object Type	Instance	Name
Read one double word starting at database address 4	Multi-state Input	8196	MSI_d_4
	Multi-state Output		MSO_d_4
	Multi-state Value		MSV_d_4
Write one double word starting	Multi-state Output	12284	MSO_d_4092



Description	Object Type	Instance	Name
at database address 4092	Multi-state Value		MSV_d_4092

Note that this example assumes  $DB_{Size} = 4096$

## 1.6.2 Analog Input Object Settings

### **Object Name**

The name of the BACnet object. Enter a string of between 1 and 16 characters in length. All object names must be unique within a node.

### **Instance**

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

### **Database Address**

Defines the database address where the BACnet object's present value will reside. The configuration studio will not allow entry of a database address that will cause the object's present value to run past the end of the database. The highest valid database address, therefore, depends on the designated Data Type of the object.

### **Data Type**

Specifies how the object's present value will be stored in the database. This defines how many bytes will be allocated, whether the value should be treated as signed or unsigned, and whether the value should be interpreted as an integer or a floating point number. Select the desired data type from this dropdown menu.

### **Multiplier**

The amount that associated network values are scaled by prior to being stored into the database or after being retrieved from the database. Upon retrieval from the database, raw data is multiplied by the multiplier to produce a network value (to send to a client.)

### **Units**

Select the desired units from this dropdown menu. If the desired units are not available in the dropdown menu, select "Other Units" and enter the appropriate enumerated value (as defined by the BACnet Specification) in the "Unit Value" field.

### **Unit Value**

*This field is enabled only when the "Units" selection is set to "Other Units".* Enter the appropriate enumerated value (as defined by the BACnet Specification.)

### **Default COV Increment**

Defines the default COV Increment value to use for this object for change of value detection. When COV reporting is enabled for this object, COV notifications will be generated if the Present Value property changes by the COV Increment value, or more, since the last notification. A COV Increment value of 0 indicates that any change to the Present Value property will generate a COV notification. Note that this setting is only available for devices that support COV reporting.



### 1.6.3 Analog Output Object Settings

#### **Object Name**

The name of the BACnet object. Enter a string of between 1 and 16 characters in length. All object names must be unique within a node.

#### **Instance**

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

#### **Database Address**

Defines the database address where the BACnet object's present value will reside. The configuration studio will not allow entry of a database address that will cause the object's present value to run past the end of the database. The highest valid database address, therefore, depends on the designated Data Type of the object.

#### **Data Type**

Specifies how the object's present value will be stored in the database. This defines how many bytes will be allocated, whether the value should be treated as signed or unsigned, and whether the value should be interpreted as an integer or a floating point number. Select the desired data type from this dropdown menu.

#### **Multiplier**

The amount that associated network values are scaled by prior to being stored into the database or after being retrieved from the database. Upon retrieval from the database, raw data is multiplied by the multiplier to produce a network value (to send to a client.) Similarly, network values (received from a client) are divided by the multiplier before being stored into the database.

#### **Units**

Select the desired units from this dropdown menu. If the desired units are not available in the dropdown menu, select "Other Units" and enter the appropriate enumerated value (as defined by the BACnet Specification) in the "Unit Value" field.

#### **Unit Value**

*This field is enabled only when the "Units" selection is set to "Other Units".* Enter the appropriate enumerated value (as defined by the BACnet Specification.)

#### **Default COV Increment**

Defines the default COV Increment value to use for this object for change of value detection. When COV reporting is enabled for this object, COV notifications will be generated if the Present Value property changes by the COV Increment value, or more, since the last notification. A COV Increment value of 0 indicates that any change to the Present Value property will generate a COV notification. Note that this setting is only available for devices that support COV reporting.

#### **Relinquish Default**

Defines the default value to be used for an object's present value property when all entries in the object's priority array are NULL.



## 1.6.4 Analog Value Object Settings

### **Object Name**

The name of the BACnet object. Enter a string of between 1 and 16 characters in length. All object names must be unique within a node.

### **Instance**

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

### **Database Address**

Defines the database address where the BACnet object's present value will reside. The configuration studio will not allow entry of a database address that will cause the object's present value to run past the end of the database. The highest valid database address, therefore, depends on the designated Data Type of the object.

### **Data Type**

Specifies how the object's present value will be stored in the database. This defines how many bytes will be allocated, whether the value should be treated as signed or unsigned, and whether the value should be interpreted as an integer or a floating point number. Select the desired data type from this dropdown menu.

### **Multiplier**

The amount that associated network values are scaled by prior to being stored into the database or after being retrieved from the database. Upon retrieval from the database, raw data is multiplied by the multiplier to produce a network value (to send to a client.) Similarly, network values (received from a client) are divided by the multiplier before being stored into the database.

### **Units**

Select the desired units from this dropdown menu. If the desired units are not available in the dropdown menu, select "Other Units" and enter the appropriate enumerated value (as defined by the BACnet Specification) in the "Unit Value" field.

### **Unit Value**

*This field is enabled only when the "Units" selection is set to "Other Units".* Enter the appropriate enumerated value (as defined by the BACnet Specification.)

### **Default COV Increment**

Defines the default COV Increment value to use for this object for change of value detection. When COV reporting is enabled for this object, COV notifications will be generated if the Present Value property changes by the COV Increment value, or more, since the last notification. A COV Increment value of 0 indicates that any change to the Present Value property will generate a COV notification. Note that this setting is only available for devices that support COV reporting.

### **Relinquish Default**

Defines the default value to be used for an object's present value property when all entries in the object's priority array are NULL.



## 1.6.5 Binary Input Object Settings

### **Object Name**

The name of the BACnet object. Enter a string of between 1 and 16 characters in length. All object names must be unique within a node.

### **Instance**

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

### **Database Address**

Defines the database address where the BACnet object's present value will reside.

### **Data Type**

Fixed at 8-Bit Unsigned.

### **Bitmask**

Specifies which bit(s) in the byte designated by the "Database Address" that the binary object will map to. This mechanism allows up to 8 binary objects to be simultaneously assigned to one database address (each binary object mapping to a single bit of that byte in the database). It is possible to map binary objects to multiple bits within the designated database location.

The effect of the "Bitmask" field when reading: When the present value property of a binary object is read by a BACnet client, the bitmask is used to determine the active/inactive state of the object by inspecting the value in the designated database address at the bit location(s) indicated in the bitmask. If all of the bit locations at the designated database address indicated by a checkmark in the bitmask are set, then the object's state will be returned as "active". Else, the object's state will be returned as "inactive". This resultant state is reversed prior to being placed on the network if the object's "Polarity" is set to "Reverse".

### **Active Text**

Specifies the description of the object's "active" state. Enter a string of up to 8 characters in length. This field is optional and may be left blank.

### **Inactive Text**

Specifies the description of the object's "inactive" state. Enter a string of up to 8 characters in length. This field is optional and may be left blank.

### **Polarity**

Indicates the relationship between the physical state of the object (as stored in the internal database) and the logical state represented by the object's present value property. If the physical state is active high, select "Normal" from this dropdown menu. If the physical state is active low, select "Reverse" from this dropdown menu. For further detail, refer to the "Bitmask" behavioral description.

## 1.6.6 Binary Output Object Settings

### **Object Name**

The name of the BACnet object. Enter a string of between 1 and 16 characters in length. All object names must be unique within a node.



### **Instance**

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

### **Database Address**

Defines the database address where the BACnet object's present value will reside.

### **Data Type**

Fixed at 8-Bit Unsigned.

### **Bitmask**

Specifies which bit(s) in the byte designated by the "Database Address" that the binary object will map to. This mechanism allows up to 8 binary objects to be simultaneously assigned to one database address (each binary object mapping to a single bit of that byte in the database). It is possible to map binary objects to multiple bits within the designated database location. Such a configuration allows (for example) the modification of multiple selected database bits via a single binary output.

The effect of the "Bitmask" field when writing: When the present value property of a binary object is set to "active" by a BACnet client, then the bit(s) in the designated database address indicated by a checkmark in the bitmask are set. Similarly, when the present value property of the object is set to "inactive", then the bit(s) in the designated database address indicated by a checkmark in the bitmask are cleared. This setting/clearing behavior is reversed if the object's "Polarity" is set to "Reverse".

The effect of the "Bitmask" field when reading: When the present value property of a binary object is read by a BACnet client, the bitmask is used to determine the active/inactive state of the object by inspecting the value in the designated database address at the bit location(s) indicated in the bitmask. If all of the bit locations at the designated database address indicated by a checkmark in the bitmask are set, then the object's state will be returned as "active". Else, the object's state will be returned as "inactive". This resultant state is reversed prior to being placed on the network if the object's "Polarity" is set to "Reverse".

### **Active Text**

Specifies the description of the object's "active" state. Enter a string of up to 8 characters in length. This field is optional and may be left blank.

### **Inactive Text**

Specifies the description of the object's "inactive" state. Enter a string of up to 8 characters in length. This field is optional and may be left blank.

### **Polarity**

Indicates the relationship between the physical state of the object (as stored in the internal database) and the logical state represented by the object's present value property. If the physical state is active high, select "Normal" from this dropdown menu. If the physical state is active low, select "Reverse" from this dropdown menu. For further detail, refer to the "Bitmask" behavioral description.



### **Relinquish Default**

Defines the default value to be used for an object's present value property when all entries in the object's priority array are NULL.

## **1.6.7 Binary Value Object Settings**

### **Object Name**

The name of the BACnet object. Enter a string of between 1 and 16 characters in length. All object names must be unique within a node.

### **Instance**

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

### **Database Address**

Defines the database address where the BACnet object's present value will reside.

### **Data Type**

Fixed at 8-Bit Unsigned.

### **Bitmask**

Specifies which bit(s) in the byte designated by the "Database Address" that the binary object will map to. This mechanism allows up to 8 binary objects to be simultaneously assigned to one database address (each binary object mapping to a single bit of that byte in the database). It is possible to map binary objects to multiple bits within the designated database location. Such a configuration allows (for example) the modification of multiple selected database bits via a single binary value.

The effect of the "Bitmask" field when writing: When the present value property of a binary object is set to "active" by a BACnet client, then the bit(s) in the designated database address indicated by a checkmark in the bitmask are set. Similarly, when the present value property of the object is set to "inactive", then the bit(s) in the designated database address indicated by a checkmark in the bitmask are cleared.

The effect of the "Bitmask" field when reading: When the present value property of a binary object is read by a BACnet client, the bitmask is used to determine the active/inactive state of the object by inspecting the value in the designated database address at the bit location(s) indicated in the bitmask. If all of the bit locations at the designated database address indicated by a checkmark in the bitmask are set, then the object's state will be returned as "active". Else, the object's state will be returned as "inactive".

### **Active Text**

Specifies the description of the object's "active" state. Enter a string of up to 8 characters in length. This field is optional and may be left blank.

### **Inactive Text**

Specifies the description of the object's "inactive" state. Enter a string of up to 8 characters in length. This field is optional and may be left blank.



## **Relinquish Default**

Defines the default value to be used for an object's present value property when all entries in the object's priority array are NULL.

## **1.6.8 Multi-state Input Object Settings**

### **Object Name**

The name of the BACnet object. Enter a string of between 1 and 16 characters in length. All object names must be unique within a node.

### **Instance**

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFE).

### **Database Address**

Defines the database address where the BACnet object's present value will reside. The configuration studio will not allow entry of a database address that will cause the object's present value to run past the end of the database. The highest valid database address, therefore, depends on the designated Data Type of the object.

### **Data Type**

Specifies how the object's present value will be stored in the database. This defines how many bytes will be allocated for the value. Select the desired data type from this dropdown menu.

### **Automatic Number of States**

When enabled, this option automatically sets the Number of States setting to the largest value allowed by the selected Data Type.

### **Number of States**

Defines the number of states that the object's present value property may have. The present value of a multi-state object is restricted to a range of 1...Number of States.

### **Offset by One**

When enabled, this option automatically offsets the object's present value property by one compared to the object's value in the database. This option is useful when mapping a multi-state object to an enumeration that starts at zero, as zero is an invalid value for a multi-state object's present value property.

The effect of the "Offset by One" field when reading: When the present value property of a multi-state object is read by a BACnet client, the raw data retrieved from the device's database is incremented by one to produce a network value.

## **1.6.9 Multi-state Output Object Settings**

### **Object Name**

The name of the BACnet object. Enter a string of between 1 and 16 characters in length. All object names must be unique within a node.



### **Instance**

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFFFE).

### **Database Address**

Defines the database address where the BACnet object's present value will reside. The configuration studio will not allow entry of a database address that will cause the object's present value to run past the end of the database. The highest valid database address, therefore, depends on the designated Data Type of the object.

### **Data Type**

Specifies how the object's present value will be stored in the database. This defines how many bytes will be allocated for the value. Select the desired data type from this dropdown menu.

### **Automatic Number of States**

When enabled, this option automatically sets the Number of States setting to the largest value allowed by the selected Data Type.

### **Number of States**

Defines the number of states that the object's present value property may have. The present value of a multi-state object is restricted to a range of 1...Number of States.

### **Offset by One**

When enabled, this option automatically offsets the object's present value property by one compared to the object's value in the database. This option is useful when mapping a multi-state object to an enumeration that starts at zero, as zero is an invalid value for a multi-state object's present value property.

The effect of the "Offset by One" field when writing: When the present value property of a multi-state object is written by a BACnet client, the network value is decremented by one before being stored into the device's database.

The effect of the "Offset by One" field when reading: When the present value property of a multi-state object is read by a BACnet client, the raw data retrieved from the device's database is incremented by one to produce a network value.

### **Relinquish Default**

Defines the default value to be used for an object's present value property when all entries in the object's priority array are NULL.

## **1.6.10 Multi-state Value Object Settings**

### **Object Name**

The name of the BACnet object. Enter a string of between 1 and 16 characters in length. All object names must be unique within a node.

### **Instance**

The BACnet object's instance number. Enter a value between 0...4194302 (0x0...0x3FFFFFFE).

### **Database Address**

Defines the database address where the BACnet object's present value will reside. The configuration studio will not allow entry of a database address that will cause the object's present value to run past the end of the database. The highest valid database address, therefore, depends on the designated Data Type of the object.

### **Data Type**

Specifies how the object's present value will be stored in the database. This defines how many bytes will be allocated for the value. Select the desired data type from this dropdown menu.

### **Automatic Number of States**

When enabled, this option automatically sets the Number of States setting to the largest value allowed by the selected Data Type.

### **Number of States**

Defines the number of states that the object's present value property may have. The present value of a multi-state object is restricted to a range of 1...Number of States.

### **Offset by One**

When enabled, this option automatically offsets the object's present value property by one compared to the object's value in the database. This option is useful when mapping a multi-state object to an enumeration that starts at zero, as zero is an invalid value for a multi-state object's present value property.

The effect of the "Offset by One" field when writing: When the present value property of a multi-state object is written by a BACnet client, the network value is decremented by one before being stored into the device's database.

The effect of the "Offset by One" field when reading: When the present value property of a multi-state object is read by a BACnet client, the raw data retrieved from the device's database is incremented by one to produce a network value.

### **Relinquish Default**

Defines the default value to be used for an object's present value property when all entries in the object's priority array are NULL.

## **1.7 Relinquish Default Event (PicoPort/Mirius Only)**

Each commandable object, i.e. Output and Value objects, can optionally include a relinquish default event. This adds the ability to detect when a relinquish command has caused the object's present value to be set to the relinquish default value. The event is assigned to a byte in the database. When a relinquish command has caused the object's present value to be set to the relinquish default value, the event database location is set to a value of 1.

### **Relinquish Event Database Address**

Specifies the database address to use for detecting that a relinquish command has caused the object's present value to be set to the relinquish default value.



---

**INDUSTRIAL CONTROL COMMUNICATIONS, INC.**

230 Horizon Drive, Suite 100  
Verona, WI USA 53593  
Tel: [608] 831-1255

<http://www.iccdesigns.com>

Printed in U.S.A