FieldServer

FS-8700-16 BACnet PTP – Serial

Driver Manual

(Supplement to the FieldServer Instruction Manual)

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after April 2020.

Driver Revision: 2.01

Document Revision: 1.B

Technical Support

Please call us for any technical support needs related to the FieldServer product.

MSA Safety
1991 Tarob Court
Milpitas, CA 95035

Website: www.sierramonitor.com

U.S. Support Information:

+1 408 964-4443
+1 800 727-4377

Email: smc-support@msasafety.com

EMEA Support Information:

+31 33 808 0590

Email: smc-support.emea@msasafety.com

TABLE OF CONTENTS

1 BACNET PTP DESCRIPTION

The BACnet PTP driver allows the FieldServer to transfer data to and from devices using the BACnet protocol over a serial RS-232 physical layer. The FieldServer can emulate either a Server or Client.

All information in a BACnet system is represented in terms of objects. The Object_Identifier is a 32-bit code that identifies the type of Object (also identified by the Object_Type Property) and its "Instance" number, which together uniquely identify the Object within its BACnet device. Theoretically, a BACnet device could have over four million Objects of a particular type. The Object_Name is a text string, which has a unique capability. BACnet devices may broadcast queries for devices that contain Objects with a specific Object_Name. This can greatly simplify project setup.

BACnet requires one Device Object to be present in every BACnet device. The Device Object makes information about the device and its capabilities available to other devices on the networks. Before one BACnet device starts control-related communications with another, it needs to obtain some of the information presented by the other device's Device Object. Unlike other Objects, the Device Object's Instance number must be unique across the entire BACnet internetwork because it is used to uniquely identify the BACnet devices. It may be used to conveniently identify the BACnet device from other devices during installation.

Standard object types are used to hold real time data and other information. Each Object Type is referenced by a number, for example 0 represents an Analog Input.

Each Object consists of a number of prescribed properties, the main property being the Present_Value. Objects are monitored and controlled through their properties. The Analog Input Object is representative of the Objects involved directly with control elements and many of its Properties reflect this.

The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer.

2 DRIVER SCOPE OF SUPPLY

2.1 Supplied by MSA Safety

Part #	Description
FS-8917-03	Connector adapter – DB9M to DCE, RTS/CTS, DSR/DTR

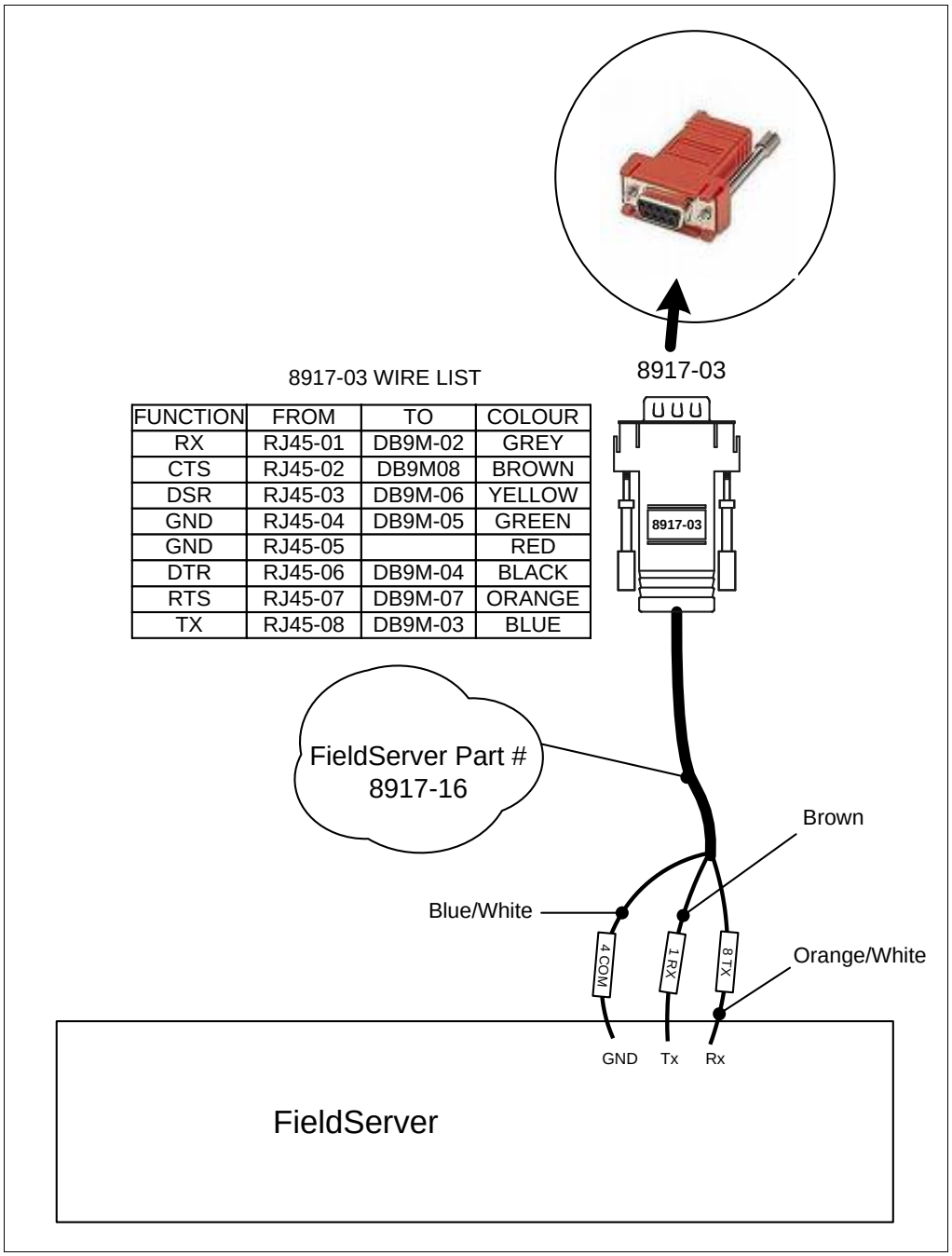
2.2 Provided by supplier of 3rd party equipment

Part #	Description
	BACnet Controller

3 HARDWARE CONNECTIONS

It is possible to connect a BACnet device to any of the eight RS-232[1] ports. These ports just need to be configured for BACnet in the configuration file.

Configure the device according to manufacturer's instructions:



[1] Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

4 DATA ARRAY PARAMETERS

Data Arrays are “protocol neutral” data buffers for storage of data to be passed between protocols. It is necessary to declare the data format of each of the Data Arrays to facilitate correct storage of the relevant data.

Section Title
Data_Arrays

Column Title	**Function**	**Legal Values**
Data_Array_Name	Provide name for Data Array.	Up to 15 alphanumeric characters
Data_Array_Format	Provide data format. Each Data Array can only take on one format.	Float, Bit, Byte, Uint16, Uint32, Sint16, Sint32
Data_Array_Length	Number of Data Objects. Must be larger than the data storage area required by the Map Descriptors for the data being placed in this array.	1-10000

Example

```
//    Data Arrays

Data_Arrays
Data_Array_Name    , Data_Array_Format    , Data_Array_Length
DA_AI_01           , Float                , 200
DA_AO_01           , Float                , 200
DA_DI_01           , Bit                  , 200
DA_DO_01           , Bit                  , 200
```

5 CONFIGURING THE FIELDSERVER AS A BACNET CLIENT

For detailed information on FieldServer configuration, refer to the FieldServer Instruction Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (see ".csv" sample files provided with the FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a BACnet PTP Server.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for BACnet communications, the driver independent FieldServer buffers need to be declared in the "Data Arrays" section, the destination device addresses need to be declared in the "Client Side Nodes" section, and the data required from the Servers needs to be mapped in the "Client Side Map Descriptors" section. Details on how to do this can be found below.

NOTE: In the following tables, * indicates an optional parameter and bold legal values are default.

5.1 Setting the FieldServer Node_ID

Section Title		
FieldServer		
Column Title	**Function**	**Legal Values**
System_Node_ID	Configure Node_ID of FieldServer.	1-255

Example

```
//  FieldServer

FieldServer
Title              , System_Node_ID
BACnet_PTP         , 11
```

5.2 Client Side Connection Parameters

Section Title		
Connections		
Column Title	**Function**	**Legal Values**
Port	Port Name.	P1-P2[2]
Baud*	Specify baud rate.	110; 300; 600; 1200; 2400; 4800; **9600**; 19200; 38400; 57600; 115000
Parity*	Specify parity.	Odd, Even, **None**
Data_Bits*	Specify data bits.	7, **8**
Stop_Bits*	Specify stop bits.	**1**,2
Protocol	Specify Protocol Used.	BACnet_PTP

Example

```
//  Client Side Connections

Connections
Port            , Baud   , Parity   , Data_Bits   , Stop_Bits   , Protocol
P2              , 9600   , Even     , 7           , 1           , BACnet_PTP
```

[2] Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

5.3 Client Side Node Parameters

Section Title
Nodes

Column Title	Function	Legal Values
Node_Name	Provide name for Node.	Up to 32 alphanumeric characters
Node_ID	Specify the BACnet Device object instance.	1 - 16777215
Protocol	Specify protocol used.	BACnet_PTP
Port	Specify port.	P1-P2
Retries*	Number of timeouts before the Node goes offline.	0,1,2,**3**

Example

```
//    Client Side Nodes

Nodes
Node_Name    , Node_ID    , Protocol          , Port    , Retries
DEV_1        , 1          , BACnet_PTP        , P1      , 2
```

5.4 Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor. This is used for Object_Name Property. Refer to **Appendix A.1**.	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer.	One of the Data Array names from **Section 4**
Data_Array_Offset	Starting location in Data Array.	0 to (Data_Array_Length-1) as specified in **Section 4**
Function	Function of Client Map Descriptor.	Rdbc, Wrbc, Wrbx, Arcs

5.4.1 Client Side Connection Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from.	One of the Node_Names specified in **Section 5.3**
Object_Type (Alias = Data_Type)	Type of object. Refer to **Appendix D.1** for more information.	AI, AO, AV, BI, BO, BV, MI, MO, MV, NC
Object_Instance (Alias = Address)	Instance of the object on the device.	0, 1, 2, 3, …4194303
Property	The BACnet property to be read.	Refer to **Appendix D.2**
Data_Array_Low_Scale*	Scaling zero in Data Array.	-32767 to 32767, **0**
Data_Array_High_Scale*	Scaling max in Data Array.	-32767 to 32767, **100**
Node_Low_Scale*	Scaling zero in Connected Node.	-32767 to 32767, **0**
Node_High_Scale*	Scaling max in Connected Node.	-32767 to 32767, **100**
Length*	Used to create an array of sequential Object_Instances on an Object_Type.	1 to max point count of the FieldServer, **1**
Array_Index*	When referencing Multistate properties, allows the user to specify the index of the property to be read. If 1 is specified, the first one will be read, if 2 is specified, the second will be read, etc. If 0 is specified, the driver will return the total number (count) of array items linked to the property. If the parameter is not specified, a list of all items will be returned. Refer to **Appendix A.3**.	0 to max number of array items in the BACnet Property, **-**
Write_Priority*	Allows the driver to specify the write priority used to write an output.	1..16, **16**
Linked_Map_Descriptors*	This parameter allows the linking of Map Descriptors to an active read or write Map Descriptor in order to construct a ReadPropertyMultiple or WritePropertyMultiple transaction. Refer to **Appendix A.9**.	The name of a previously defined active Map Descriptor, -
Length*	When the optional Length parameter is set to a value N, the Map Descriptor will include N consecutive instances of the specified Object_Type. For example, if a Map Descriptor specifies Object_Type, Address 4 and Length 3, this means that the AO instances 4, 5 and 6 will be included in the read or write transaction. The corresponding Data Array values are in consecutive positions in the Data Array, starting at the specified Data_Array_Offset.	Any positive integer that falls between the range of the Data_Array_Offset and the Data_Array_Length as specified in **Section 4**, - Refer also to **Appendix A.9**

5.4.2 Timing Parameters

Column Title	Function	Legal Values
Scan_Interval*	Rate at which data is polled	0-32000, **2s**

5.4.3 Map Descriptor Example

```
//    Client Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name   , Data_Type   , Object_Instance   , Property   , Scan_Interval
CMD_AI_01             , DA_AI_01          , 0                   , Rdbc       , DEV_1       , AI          , 1                 , Present_Value , 20.000s
CMD_AI_02             , DA_AI_01          , 1                   , Rdbc       , DEV_1       , AI          , 2                 , Present_Value , 20.000s
CMD_AI_03             , DA_AI_01          , 2                   , Rdbc       , DEV_1       , AI          , 3                 , Present_Value , 20.000s
CMD_AO_01             , DA_AO_01          , 0                   , Rdbc       , DEV_1       , AO          , 1                 , Present_Value , 30.000s
CMD_AO_02             , DA_AO_01          , 1                   , Rdbc       , DEV_1       , AO          , 2                 , Present_Value , 30.000s
CMD_AO_03             , DA_AO_01          , 2                   , Rdbc       , DEV_1       , AO          , 3                 , Present_Value , 30.000s


Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name   , Data_Type   , Object_Instance   , Property   , Scan_Interval
CMD_DI_01             , DA_DI_01          , 0                   , Rdbc       , DEV_1       , BI          , 1                 , Present_Value , 15.000s
CMD_DI_02             , DA_DI_01          , 1                   , Rdbc       , DEV_1       , BI          , 2                 , Present_Value , 15.000s
CMD_DI_03             , DA_DI_01          , 2                   , Rdbc       , DEV_1       , BI          , 3                 , Present_Value , 15.000s
CMD_DO_01             , DA_DO_01          , 0                   , Rdbc       , DEV_1       , BO          , 1                 , Present_Value , 30.000s
CMD_DO_02             , DA_DO_01          , 1                   , Rdbc       , DEV_1       , BO          , 2                 , Present_Value , 30.000s
CMD_DO_03             , DA_DO_01          , 2                   , Rdbc       , DEV_1       , BO          , 3                 , Present_Value , 30.000s
```

6 CONFIGURING THE FIELDSERVER AS A BACNET SERVER

For detailed information on FieldServer configuration, refer to the FieldServer Configuration Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (see ".csv" sample files provided with the FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a BACnet PTP Client.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for BACnet communications, the driver independent FieldServer buffers need to be declared in the "Data Arrays" section, the FieldServer virtual node(s) needs to be declared in the "Server Side Nodes" section, and the data to be provided to the clients needs to be mapped in the "Server Side Map Descriptors" section. Details on how to do this can be found below.

NOTE: In the tables below, * indicates an optional parameter, with the bold legal value as default.

6.1 Driver Specific FieldServer Parameters

Section Title
Bridge

Column Title	Function	Legal Values
Title	FieldServer name.	Text
Network_Number*	Specify a unique network number if there are multiple virtual Server nodes. Refer to **Appendix A.2**.	1-65535 **5**

Example

```
//  FieldServer Driver specific parameters

Bridge
Title
BACnet Server
```

6.2 Server Side Connection Parameters

Section Title
Connections

Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer.	P1-P2
Baud*	Specify baud rate.	110; 300; 600; 1200; 2400; 4800; **9600**; 19200; 38400; 57600; 115000
Parity*	Specify parity.	**None,** Odd, Even
Data_Bits*	Specify data bits.	7**, 8**
Stop_Bits*	Specify stop bits.	**1**, 2
Protocol	Specify protocol used.	BACnet_PTP
Reliability_Option*	The Reliability_Option parameter selects the rules by which the BACnet Server determines the Reliability property of a BACnet object whose present value is read from a Remote Server Node. Refer to **Appendix A.8**.	Track_Object_Status, **Track_Node_Status,** Always_Reliable

Example

```
//    Server Side Connections

Connections
Port            , Baud    , Parity   , Data_Bits   , Stop_Bits   , Protocol
P1              , 9600    , Even     , 7           , 1           , BACnet_PTP
```

6.3 Server Side Node Parameters

Section Title
Nodes

Column Title	Function	Legal Values
Node_Name[3]	Provide name for node.	Up to 32 alphanumeric characters
Node_ID	BACnet station address of physical Server node.	1 - 16777215
Protocol	Specify protocol used.	BACnet_PTP
Node_Option	Enable or disable COV for this node.	**COV_Enable**, COV_Disable

Example

```
//    Server Side Nodes

Nodes
Node_Name       , Node_ID    , Protocol [4]
Virtual_DEV_11  , 11         , BACnet_PTP
```

6.4 Server Side Map Descriptor Parameters

6.4.1 FieldServer Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor. This is used for Object_Name Property. Refer to **Appendix A.1**.	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer.	One of the Data Array names from **Section 4**
Data_Array_Offset	Starting location in Data Array.	0 to (“Data_Array_Length” -1) as specified in **Section 4**
Function	Function of Server Map Descriptor.	Passive

[3] When using the Trane Tracer Summit Workstation System, this name will appear as the object name.
[4] Note that Port is not declared in the Server Side Nodes.

6.4.2 Driver Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from.	One of the Node Names specified in "Server Side Node Descriptors"
Object_Type (Alias = Data_Type)	Data type in Controller.	AI, AO, AV, BI, BO, BV, MI, MO, MV, NC. Refer to **Appendix D.1**.
Object_Instance (Alias = Address)	Instance of the Object on the Device.	0, 1, 2, 3, …4194303
Units*	The object units.	Refer to **Appendix D.3**, **m2**
Data_Array_Low_Scale*	Scaling zero in Data Array.	-32767 to 32767, **0**
Data_Array_High_Scale*	Scaling max in Data Array.	-32767 to 32767, **100**
Node_Low_Scale*	Scaling zero in Connected Node.	-32767 to 32767, **0**
Node_High_Scale*	Scaling max in Connected Node.	-32767 to 32767, **100**
Active_Text[5]*	Specify the Active Text property of the Object.	Any text string of length up to 40 characters, **Active**
Inactive_Text[5]*	Specify the Inactive Text property of the Object.	Any text string of length up to 40 characters, **Inactive**
Relinquish_Default[5]	Specify the value to be returned as Present_Value on startup or when control is relinquished. Must be specified for outputs. Refer to **Appendix D.2**.	Any Float value
Length*	Used to create an array of sequential Object_Instances on an Object_Type.	1 to max point count of the FieldServer, **1**
Notification_Class*	Specify the Mapdescriptor_Name of the Notification_Class Object that manages Intrinsic Reporting for this Map Descriptor. Refer to **Appendix A.6**.	One of the configured Mapdescriptor_Names of type NC (Notification Class). Objects are not available to any NC if left out.
Ack_Required*	For a Notification_Class Object, specify whether EventNotifications require a user Acknowledgement. Refer to **Appendix A.6**.	Yes, **No**
COV_Increment*	For a Server Map Descriptor of type AO or AI, initialize the COV_Increment property. Refer to **Appendix A.6**.	Any Float value, **0**
Input_alarm_State*	For binary points the user can specify which of the states (0 or 1) to regard as an alarm state.	0, **1**
Description*	Specify the object's description property. E.G. Room Temp – Refer to **Appendix A.7**.	Any text string of length up to 40 characters, if not configured, defaults to Object_Name.

[5] See the BACnet DFS to determine if a particular object supports this property.

6.4.3 Map Descriptor Example

```
//    Server Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name       , Object_Type   , Object_Instance   , Units
SMD_AI_01             , DA_AI_01          , 0                   , Passive    , Virtual_DEV_11   , AI            , 1                 , Degrees-Fahrenheit
SMD_AI_02             , DA_AI_01          , 1                   , Passive    , Virtual_DEV_11   , AI            , 2                 , Degrees-Fahrenheit
SMD_AI_03             , DA_AI_01          , 2                   , Passive    , Virtual_DEV_11   , AI            , 3                 , Degrees-Fahrenheit
SMD_AO_01             , DA_AO_01          , 0                   , Passive    , Virtual_DEV_11   , AO            , 1                 , percent-relative-humidity
SMD_AO_02             , DA_AO_01          , 1                   , Passive    , Virtual_DEV_11   , AO            , 2                 , percent-relative-humidity
SMD_AO_03             , DA_AO_01          , 2                   , Passive    , Virtual_DEV_11   , AO            , 3                 , percent-relative-humidity

Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name       , Object_Type   , Object_Instance
SMD_DI_01             , DA_DI_01          , 0                   , Passive    , Virtual_DEV_11   , BI            , 1
SMD_DI_02             , DA_DI_01          , 1                   , Passive    , Virtual_DEV_11   , BI            , 2
SMD_DI_03             , DA_DI_01          , 2                   , Passive    , Virtual_DEV_11   , BI            , 3
SMD_DO_01             , DA_DO_01          , 0                   , Passive    , Virtual_DEV_11   , BO            , 1
SMD_DO_02             , DA_DO_01          , 1                   , Passive    , Virtual_DEV_11   , BO            , 2
SMD_DO_03             , DA_DO_01          , 2                   , Passive    , Virtual_DEV_11   , BO            , 3
```

Appendix A. Useful Features

Appendix A.1. BACnet Object Names

When an external BACnet Client builds a list of Object Names, the BACnet Server Map Descriptor names determine the BACnet Object Name. If the Map Descriptor length is greater than 1, the Object Name will be suffixed with the index into the Map Descriptor. For example, if the Map Descriptor name is SMD_AI_01 and the length 3, then the Object Names will be SMD_AI_01[0], SMD_AI_01[1] and SMD_AI_01[2].

Appendix A.2. Network Number

If multiple BACnet Nodes are specified on the Server side, the FieldServer automatically defaults to virtual operation. In effect, the FieldServer simulates a software router. BACnet identifies this FieldServer using a combination of its network number and IP/MAC address. The default Network number of a FieldServer is 5. If there is more than one FieldServer on a BACnet intranet with multiple nodes, the default network number of the additional FieldServers will need to be changed and a unique number allocated to each.

A unique network number will need to be assigned if *both* of the following conditions are true:

- The FieldServer has multiple BACnet Server nodes.
- There is more than one FieldServer on a network which includes multiple BACnet nodes.

To override the FieldServer's default network number 5 include the following in the configuration file:

Section Title		
Bridge		
Column Title	**Function**	**Legal Values**
Title	FieldServer name.	Text
Network_Number*	Specify a unique network number if there are multiple virtual Server nodes.	1-65534, **5**

Example

```
//  FieldServer Driver specific parameters

Bridge
Title                , Network_Number
BACnet Server        , 6
```

NOTE: While it is theoretically possible to have up to 65535 virtual nodes, it is recommended that a maximum of 32 is configured. If an application requires the configuration of more than 32 virtual nodes, consult FieldServer technical support.

Appendix A.3. Accessing Data from BACnet Properties Comprising Arrays of Values

Some BACnet Object properties (e.g. Priority_Array) are arrays of values (the Priority_Array property is an array of 16 values). In order to read a specific array entry, the Array_Index must be specified in the Map Descriptor. Array_Index is a Client Side Map Descriptor function

The following example shows a configuration that will read the Priority_Array value at Array_Index 7, belonging to Analog Output 1.

```
Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name
CMD_AI_01             , DA_AI_01          , 0                  , Rdbc       , DEV_01
```

```
, Data_Type   , Object_ID   , Property        , Array_Index
, AO          , 1           , Priority_Array  , 7
```

Details of the relevant BACnet properties and their associated arrays can be found in the BACnet Protocol Spec.

Appendix A.4. FieldServer Implementation of BACnet Priority Arrays

When BACnet Output objects are written to the Server side of the FieldServer, an associated write priority is given to each write value. When the FieldServer receives the write value, it stores it to the Map Descriptor Priority Array Table at the specified priority. The Priority Array Table is then scanned and the value with the highest priority is stored to the Data Array location specified by the Map Descriptor.

When a Write "Relinquished" command is received, the value is removed from the Priority Array Table and the next highest value from the Priority Array Table is stored to the Data Array.

If all values have been "Relinquished" from the Priority Array Table, then the Map Descriptors "Relinquish Default" value will be stored to the Data Array.

Appendix A.4.1. Accessing Priority Array Information

The Priority Array table and its "In_Use" (or Not Relinquished) state are stored internally to every Map Descriptor and cannot be accessed directly. The information can be accessed indirectly by specifying the following Data Arrays which will maintain an exact copy of the Priority Array Table for the Map Descriptor.

Section Title		
Map_Descriptors		
Column Title	**Function**	**Legal Values**
DA_Pri_Array	Name of the Data Array that contains the Priority Array Table. Location 0 is the Relinquish Default value and locations 1 to 16 the different entries of the Priority Array Table.	Up to 16 alphanumeric characters
DA_Pri_Array_Offset*	Starting location in Data Array.	1-65535, **0**
DA_Pri_In_Use	Name of Data Array that indicates if a particular Priority Value is in use. Location 0 indicates whether the Relinquish Default has been set and locations 1 to 16 indicate whether the index is in use (1), or Relinquished (0).	Up to 16 alphanumeric characters
DA_Pri_In_Use_Offset*	Starting location in Data Array.	1-65535, **0**

Example

```
//  Analog Output Map_Descriptor for testing Priority Arrays
Map_Descriptors
Map_Descriptor_Name  , Data_Type  , Object_ID  , Function  , Data_Array_Name  , Data_Array_Index  , Node_Name  , Length
CMD_AOP_1            , AO         , 1          , Passive   , DA_OUT           , 0                 , N1 11       , 1
```

```
, Relinquish_default  , DA_Pri_Array     , DA_Pri_Array_Offset  , DA_Pri_In_Use     , DA_Pri_In_Use_Offset
, 40.56               , DA_Pri_Array_1   , 0                    , DA_Pri_in_use_1   , 0
```

Appendix A.5. BACnet State Text Preload

BACnet Multistate Objects have a State_Text property. This property is defined as an array of character strings representing descriptions of all possible states of the Present_Value. The number of descriptions matches the number of states defined in the Number_Of_States property. The Present_Value, interpreted as an integer, serves as an index into the array.

When Multistate Objects are configured on a BACnet Server it is necessary to define the State_Text property. This section illustrates how to define the State_Text character strings and how to associate these definitions with Multistate Server Map Descriptors.

The maximum permitted length of any State_Text string is 50 characters.

Appendix A.5.1. Method 1 – Using an Offset/User Table

```
//set up a look up table

Offset_Table
Offset_Table_Name     , Table_String        , Table_Index_Value
FIRE_ALRM_TEXT        , SYSTEM READY        , 1
FIRE_ALRM_TEXT        , ALARM               , 2
FIRE_ALRM_TEXT        , MAINTENANCE         , 3
FIRE_ALRM_TEXT        , OFF-LINE            , 4
FIRE_ALRM_TEXT        , IN SERVICE          , 5
FIRE_ALRM_TEXT        , OTHER               , 6
```

Valid values are 1 to 16.

```
Data_Arrays
Data_Array_Name   , Data_Format   , Data_Array_Length
DA_MI_01          , UINT16        , 100
```

```
Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name
CMD_MI_01             , DA_MI_01          , 0                   , Passive    , N1 11

, Data_Type   , Object_ID   , Length   , State_Text_Array
, MI          , 1           , 1        , Fire_Alrm_Text
```

Appendix A.5.2. Method 2 – Using a Single Data Array

```
Data_Arrays
Data_Array_Name   , Data_Format   , Data_Array_Length
DA_MI_01          , UINT16        , 100
DA_STATE_TXT      , BYTE          , 200
```

```
Preloads
Data_Array_Name   , Preload_Data_Value                  , Preload_Data_Format   , Preload_Data_Index
DA_STATE_TXT      , MyState1 MyState2 MyState3          , String                , 0
                    MyState4 MyState5 MyState6
```

```
Map_Descriptors
Map_Descriptor_Name   , Data_Type   , Object_ID   , Function   , Data_Array_Name   , Node_Name   , Length   , State_Text_Array
CMD_MI_01             , MI          , 1           , Passive    , DA_MI_01          , N1 11       , 1        , Da_State_Txt
```

Appendix A.5.3. Using Intrinsic Reporting for a Multistate Value

To use Intrinsic Reporting for a Multistate value, it is necessary to classify each of the states as either Normal, Alarm or Trouble. This is done by adding another column to the Offset_Table, called Table_User_Value. Each state is then classified by inserting one of the following values in its row:

0 = normal
1 = alarm
2 = fault

```
//set up a look up table

Offset_Table
Offset_Table_Name     , Table_String      , Table_Index_Value   , Table_User_Value
FIRE_ALRM_TEXT        , SYSTEM READY      , 1                   , 0
FIRE_ALRM_TEXT        , ALARM             , 2                   , 1
FIRE_ALRM_TEXT        , MAINTENANCE       , 3                   , 2
FIRE_ALRM_TEXT        , OFF-LINE          , 4                   , 2
FIRE_ALRM_TEXT        , IN SERVICE        , 5                   , 0
FIRE_ALRM_TEXT        , OTHER             , 6                   , 2
```

NOTE: The state value (Table_Index_Value) is an enumerated value between 1 and 16. Zero is not a valid value, but since many client side configurations will wake up with values of zero, the BACnet Server will treat a value of zero as a normal value (i.e. not as an alarm or fault value).

Appendix A.6. COV and Intrinsic Reporting

The COV (Change of Value) and Intrinsic Reporting services are two distinct ways in which point values can be reported to a client workstation as they change, i.e. in an event-driven opposed to a polling method. This can increase performance dramatically compared to polling method alone. It also reduces network traffic significantly.

The services are suited to different purposes:

- COV is suited to value updates. On analog points the sensitivity can be set using the COV_Increment property. Only changes larger than the COV_Increment value will be reported.
- Intrinsic Reporting is used for alarming. It is implemented via Notification_Class objects, which can receive subscriptions from client workstations that add themselves to the RecipientList property of a Notification_Class object. Notifications are done using ConfirmedEventNotification or UnconfirmedEventNotification. Intrinsic Reporting also allows for alarms to be acknowledged (using the AcknowledgeAlarm service) and for all subscribed client workstations to be notified of alarm acknowledgements (using EventNotifications of type ACK_NOTIFICATION).

Appendix A.6.1. Notes on COV Configuration

- No special configuration entries are needed to enable COVs. The service is enabled by default for all protocols except MSTP. Most client workstations will automatically subscribe to all points once they discover that the FieldServer supports COV services. This only applies to BACnet Objects - Property subscribes are not supported by the FieldServer. COV only works for BACnet Map Descriptors with length set to 1. If length is not specified, then it defaults to 1, so this is only a problem where length has been specified as greater than 1.
- The Node_Option parameter can be configured to enable or disable COV. An example configuration is presented in **Appendix A.7.3**.
- For analog Server Map Descriptors, the user may optionally configure a COV_Increment value to adjust the reporting threshold. If it is not set the COV_Increment defaults to zero.
- Change of Value (COV) Notifications are generated for all data objects for which a remote client has issued a SubscribeCOV-Request. COV subscription is on a per-point basis. The SubscribeCOV-Request regulates whether Notifications are Confirmed or Unconfirmed. The remote client may also write the COV_Increment property in order to control the deadband for changes in analog values. The COV_Increment property can be initialized via the configuration file by setting the COV_Increment Map Descriptor Property. The value set by the configuration is an initial value that is loaded on startup. It would be replaced by any new value written by the Client.

Appendix A.6.2. Notes on Intrinsic Reporting Configuration:

- Intrinsic Reporting is managed by Notification Class objects. At least one Notification Class object must be configured for Intrinsic Reporting to work.
- Each Data_Object that is to be monitored by Intrinsic Reporting must be linked to a Notification Class object via the Notification_Class Map Descriptor Property. For analog points alarm limits must be set up, and for binary points, the Input_Alarm_State must be set up.
- The Notification_Class object contains properties that allow a client workstation to modify rules governing event reporting, such as Event_Type, days of week, start and end times etc.
- The RecipientLists are non-volatile, and subscriptions must be renewed on system restart.

Appendix A.7. Map Descriptor Examples

Appendix A.7.1. COV

```
//    Change of Value

Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name      , Data_Type   , Object_ID   , Ack_Required
SMD_NC_01             , DA_NC_01          , 0                   , Passive    , Virtual_Dev_11  , NC          , 01          , -
```

```
Map_Descriptors
Map_Descriptor_Name   Description     Data_Array_Name   Data_Array_Offset   Function    Node_Name       Data_Type   Object_ID   Relinquish_Default
SMD_AI_01             , Room Temp     , DA_AI_01        , 0                 , Passive   , Virtual_Dev_11  , AI         , 01        , -
```

```
State_Text_Array   , Notification_Class   , High_Alarm   , Low_Alarm   , Input_Alarm_State   , Confirmed   , COV_Increment
, -                , SMD_NC_01           , 100          , 10          , -                  , Yes         , 1.0
```

Appendix A.7.2. Intrinsic Reporting

```
//    Notification class Objects

Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name       , Object_Type   , Object_Instance   , Ack_Required
SMC_NC_01             , DA_NC_01          , 0                   , Passive    , Virtual_DEV_11  , NC            , 1                , Yes
SMC_NC_02             , DA_NC_01          , 1                   , Passive    , Virtual_DEV_11  , NC            , 2                , No

Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name       , Object_Type   , Object_Instance   , Notification_Class
SMD_AI_01             , DA_AI_01          , 0                   , Passive    , Virtual_DEV_11  , AI            , 1                , SMC_NC_01
SMD_AO_01             , DA_AO_01          , 0                   , Passive    , Virtual_DEV_11  , AO            , 1                , SMC_NC_02

Map_Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name       , Object_Type   , Object_Instance   , Notification_Class
SMD_DI_01             , DA_DI_01          , 0                   , Passive    , Virtual_DEV_11  , BI            , 1                , SMC_NC_01
SMD_DO_01             , DA_DO_01          , 0                   , Passive    , Virtual_DEV_11  , BO            , 1                , SMC_NC_02
```

Appendix A.7.3. Enable or Disable COV

```
Nodes
Node_Name        , Node_ID   , Protocol      , Node_Option
Virtual_Dev_11   , 11        , BACnet_IP    , COV_Disable
```

Appendix A.8. Reliability_Option

The Reliability_Option parameter selects the rules by which the BACnet server determines the Reliability property of a BACnet object whose present value is read from a Remote Server Node.

There are two important events that can affect the reliability of a BACnet object:

- The responsible Remote Server Node is offline, i.e. the data cannot be updated.
- A BACnet write request has been received by the BACnet Server. This causes the corresponding FieldServer data object to be expired until its value is updated by the next successful Read operation.

There are three options for translating these conditions to BACnet Reliability values:

Track_Object_Status	The Reliability property mirrors the internal data object state, i.e. if the Remote Server Node is offline or the data is expired by a BACnet write, the Reliability is marked Unreliable-Other.
Track_Node_Status (default)	The reported reliability is good unless the Remote_Server_Node is offline, in which case the Reliability is Unreliable-Other. Reliability is not affected by data expiry following a BACnet write.
Always_Reliable	The BACnet Server always reports good data reliability, regardless of the status of communications with the Remote Server Node.

Example

```
Adapters
Adapter    , Protocol     , Reliability_Option
N1         , BACnet_IP    , Track_Node_Status
```

NOTE: This setting applies to all BACnet Nodes configured for a given BACnet Connection.

Appendix A.9. Specify Read/Write PropertyMultiple Transactions with Linked Map Descriptors

The ReadPropertyMultiple (RPM) and WritePropertyMultiple (WPM) BACnet services allow many objects and attributes to be read and written in a single transaction. Since FieldServer Map Descriptors can only refer to a single object type and address range, an RPM or WPM transaction can be constructed by linking multiple Map Descriptors. This is done using the following components:

- An active read or write Map Descriptor that defines the behavior of the RPM or WPM transaction via:
  - Function, e.g. Rdbc, Arcs, Wrbc, Wrbx
  - Scan_interval
  - Timeout
  - Length
- Any number of Linked Map Descriptors specifying additional objects and data array locations to be included in the transaction. Each Linked Map Descriptor is specified with the following settings:
  - Linked_Map_Descriptor – this is a reference by name to the active Map Descriptor described above, which controls the transaction
  - Function – Passive_Client
  - Length

The number of Linked Map Descriptors is limited by the maximum message length allowed for the BACnet driver in question. When too many Map Descriptors have been linked a SEGMENTATION_NOT_SUPPORTED error message will be generated on the first poll attempt. This message will recommend splitting the RPM or WPM transaction into multiple transactions, e.g.:

```
DRV->BACnet : Linked Map Desc. "CMD_WPM" is too long.
Message Segmentation not supported.
Please split the transaction into multiple Linked Map Descriptors.
```

Example

```
Map_Descriptors
Map_Descriptor_Name  , Object_Type  , Property  , Function  , Data_Array_Name  , Data_Array_Offset  , Node_Name  , Address  , Length  , Write_Priority  , Linked_Map_Descriptor
CMD WPM              , AO           , -         , Wrbc      , DA WPM           , 0                  , N1 1       , 1        , 2       , 7               , -
```

Configure an active read or write Map Descriptor and give it a unique name. This Map Descriptor will be referred to by all other Map Descriptors forming part of the multiple property transaction.

Configure a Map Descriptor for each Object and Property. These are component Map Descriptors forming part of the composite Map Descriptor CMD WPM above.

```
Map_Descriptor_Name, Object_Type, Property       , Function       , Data_Array_Name, Data_Array_Offset, Node_Name, Address, Length, Write_Priority, Linked_Map_Descriptor
CMD WPM 1          , AO         , -              , Passive_Client, DA WPM         , 3                , N1 1     , 4      , 3     , 8             , CMD WPM
CMD WPM 2          , AO         , -              , Passive_Client, DA WPM         , 7                , N1 1     , 8      , 4     , 10            , CMD WPM
CMD WPM 3          , Device     , Max_Master     , Passive_Client, DA WPM         , 12               , N1 1     , 1      , 1     , 10            , CMD WPM
CMD WPM 4          , Device     , Max_Info_Frames, Passive_Client, DA WPM         , 13               , N1 1     , 1      , 1     , 10            , CMD WPM
CMD WPM 5          , BO         , -              , Passive_Client, DA BOP         , 0                , N1 1     , 0      , 10    , 11            , CMD WPM
```

Set the function to Passive_Client.

Set Linked_Map_Descriptor to the name of the active read or write Map Descriptor governing the transaction.

```
// ReadPropertyMultiple

Map_Descriptors
Map_Descriptor_Name  , Object_Type  , Property  , Function  , Data_Array_Name  , Data_Array_Index  , Node_Name  , Address  , Length  , Write_Priority  , Linked_Map_Descriptor
CMD RPM              , AO           , -         , Rdbc      , DA RPM           , 9                  , N1 1       , 1        , 2       , -               , -
```

```
Map_Descriptor_Name  , Object_Type  , Property       , Function       , Data_Array_Name  , Data_Array_Index  , Node_Name
CMD RPM 3            , Device       , Max_Master     , Passive_Client , DA RPM           , 0                  , N1 1
CMD RPM 4            , Device       , Max_Info_Frames, Passive_Client , DA RPM           , 1                  , N1 1
CMD RPM 2            , AO           , -              , Passive_Client , DA RPM           , 2                  , N1 1
CMD RPM 1            , AO           , -              , Passive_Client , DA RPM           , 6                  , N1 1
```

These map descriptors form part of the read transaction defined by Map Descriptor CMD RPM above.

```
, Address  , Length  , Write_Priority  , Linked_Map_Descriptor
, 1        , 1       , -               , CMD RPM
, 1        , 1       , -               , CMD RPM
, 8        , 4       , -               , CMD RPM
, 4        , 3       , -               , CMD RPM
```

NOTE:

- For the Present_Value property it is permissible to set a Map Descriptor length >1. This will cause a read of the Present_Value property of consecutive BACnet objects of the type defined by this Map Descriptor.
- The number of properties that can be read or written at once is limited by the maximum APDU message length.
- Message segmentation is not supported.

Appendix B. Troubleshooting

Appendix B.1. Debugging a BACnet Connection

- If duplicate Object_Instances are configured in the FieldServer, the second call of the Instance will overwrite the first one. This may cause a BACnet Object to be "lost."
- If the Node Name configured on the BACnet Server Side of the configuration is not being indicated as the Device Name on the BACnet SCADA system, then the FieldServer is not communicating with the SCADA system. If the Device Object's name is being indicated, but the Present_Value shows question marks, then it is likely that the Client side of the FieldServer is not communicating.
- Extra memory is required to store Map Descriptors that have the active/inactive text parameters specified. If the defaults are appropriate, do not specify these parameters. This will save memory and allow more Map Descriptors to be created.
- When a BACnet_PTP connection is established, and the connection is broken abruptly (for instance a power down), the DEVICE might not know immediately that this happened. If the FieldServer is then powered up again, it might take several seconds (up to 60 seconds) before polling will start again.

Appendix B.2. BACnet Error Response Decoding

BACnet reports errors in the following format:

```
T02> 10/22 02:57 HEXDUMP : ERROR_PDU
T02> 10/22 02:57 0x19d2d   50 97 0f 91 02 91 2a
T02> 10/22 02:57 BACnet -> Unexpected ERROR_PDU : err_class=2 err_code=42
```

These can be decoded using the following tables:

Appendix B.2.1. BACnet Error CLASS

Device	0
Object	1
Property	2
Resources	3
Security	4
Services	5

Appendix B.2.2. BACnet Error CODES for Error Class OBJECT

Other	0
Authentication failed	1
Configuration in progress	2
Device busy	3
Dynamic creation not supported	4
File access denied	5
Incompatible security levels	6
Inconsistent parameters	7
Inconsistent selection criterion	8
Invalid data type	9
Invalid file access method	10
Invalid file start position	11
Invalid operator name	12
Invalid parameter data type	13
Invalid time stamp	14
Key generation error	15
Missing required parameter	16
No objects of specified type	17
No space for object	18
No space to add list element	19
No space to write property	20
No vt sessions available	21
Property is not a list	22
Object deletion not permitted	23
Object identifier already exists	24
Operational problem	25
Password failure	26
Read access denied	27
Security not supported	28
Service request denied	29
Timeout	30
Unknown object	31
Unknown property	32
Unknown vt class	34
Unknown vt session	35
Unsupported object type	36
Value out of range	37
Vt session already closed	38
Vt session termination failure	39
Write access denied	40
Character set not supported	41
Invalid array index	42
Invalid index	42

Appendix C. Vendor Information

Appendix C.1. McQuay

McQuay Units are shipped with a default Device instance of the last 6 digits of the McQuay Serial number.

Appendix C.2. Trane

When new points are added to the FieldServer it is important to restart the Summit Workstation or BCU, otherwise these new points may not be seen by the FieldServer.

Disconnect the FieldServer from the BACnet network when transferring images to the BCU.

Connection to a Trane BCU requires the purchase of a RTS to DB9M convertor. This is available from MSA Safety (Part # FS-8917-02).

Appendix C.3. Liebert

Polling BACnet addresses that are not configured for Liebert systems may cause the connection to fail in older versions of Liebert. Please contact your Liebert supplier for more information.

Appendix C.4. Automated Logic Corporation

When an ALC module is powered up, and it does not detect valid BACnet/MSTP traffic on its MSTP port, then the module goes into a terminal mode and MSTP communications will not be initiated.

When connecting the FieldServer to an ALC BACnet/MSTP module, always start the FieldServer first. Wait until the RUN Led is flashing on the FieldServer before powering up the MSTP module.

Appendix C.5. Honeywell EBI

Honeywell EBI cannot process EventNotifications with ACK_Required set to 1. The ACK_Required property of Notification Class Map Descriptors configured for use with EBI must therefore be set to 0.

Appendix D. Reference

Appendix D.1. Object_Type Legal Values – Abbreviation Descriptions

AI	ANALOG_INPUT
AO	ANALOG_OUTPUT
AV	ANALOG_VALUE
BI	BINARY_INPUT
BO	BINARY_OUTPUT
BV	BINARY_VALUE
MI	MULTI_STATE_INPUT
MO	MULTI_STATE_OUTPUT
MV	MULTI_STATE_VALUE
NC	NOTIFICATION_CLASS_OBJECT
DEVICE	DEVICE

Appendix D.2. Property Legal Values

Legal Value	Description
Object_Identifier	This property is a numeric code that is used to identify the object. It is unique within the BACnet Device that maintains it.
Object_List	Relevant to Device Object Type. This property is a BACnetARRAY of Object_Identifiers, one Object_Identifier for each object within the device that is accessible through BACnet services. An Object_Identifier is composed of Object Type and Object Instance and must be unique within a BACnet Device, e.g. Object Type = Analog Input, Object Instance = 3.
Present_Value	This property contains the present value of the Input / Output / Value.
Object_Name	Character string providing the name of a BACnet object. The set of characters used in the Object_Name is restricted to printable characters. The Object_Name is determined by the Map_Descriptor_Name.
Description	Character string describing a BACnet object. This can be defined by the user to give additional detail about the Object.
Out_of_service	The Out_Of_Service property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the physical input that the object represents is not in service. This means that the Present_Value property is decoupled from the physical input and will not track changes to the physical input when the value of Out_Of_Service is TRUE. In addition, the Reliability property and the corresponding state of the FAULT flag of the Status_Flags property shall be decoupled from the physical input when Out_Of_Service is TRUE. While the Out_Of_Service property is TRUE, the Present_Value and Reliability properties may be changed to any value as a means of simulating specific fixed conditions or for testing purposes. Other functions that depend on the state of the Present_Value or Reliability properties shall respond to changes made to these properties while Out_Of_Service is TRUE, as if those changes had occurred in the physical input.
Event_State	The Event_State property, of type BACnetEventState, is included in order to provide a way to determine if this object has an active event state associated with it. If the object supports intrinsic reporting, then the Event_State property shall indicate the event state of the object. If the object does not support intrinsic reporting, then the value of this property shall be NORMAL. Other values: FAULT, OFF-NORMAL, HIGH-LIMIT, LOW-LIMIT, LIFE-SAFETY-ALARM.
Units	This property contains the units associated with the Present_Value property.

Legal Value	Description
Reliability	The Reliability property, of type BACnetReliability, provides an indication of whether the Present_Value or the operation of the physical input in question is "reliable" as far as the BACnet Device or operator can determine and, if not, why. The following values are supported: NO_FAULT_DETECTED, UNRELIABLE_OTHER
Priority_Array	This property relates to Output and Value Object Types and is a read only array that contains prioritized commands or NULLs in the order of decreasing priority. The highest priority (lowest array index) with a non-NULL value is the active command.
State_Text	Relevant to Multistate Object Types: This property is a BACnetARRAY of character strings representing descriptions of all possible states of the Present_Value. The number of descriptions matches the number of states defined in the Number_Of_States property. The Present_Value, interpreted as an integer, serves as an index into the array.
Number_Of_States	Relevant to Multistate Object Types: this property sets the total number of states for which descriptions will be returned as defined under the State_Text property. The number of states will be determined automatically by the largest state number used when configuring the Offset Table (refer to **Appendix A.5**).
Max_Master	Relevant to BACnet MS/TP Device Object Type: The Max_Master property, of type Unsigned, shall be present if the device is a master node on an MS/TP network. The value of Max_Master specifies the highest possible address for master nodes and shall be less than or equal to 127. If the Max_Master property is not writeable via BACnet services, its value shall be 127.
Max_Info_Frames	Relevant to BACnet MS/TP Device Object Type: The Max_Info_Frames property, of type Unsigned, shall be present if the device is a node on an MS/TP network. The value of Max_Info_Frames specifies the maximum number of information frames the node may send before it must pass the token. If Max_Info_Frames is not writable or otherwise user configurable, its value shall be 1.
Active_Text	Relevant to Binary Object Types: This property, of type CharacterString, characterizes the intended effect of the ACTIVE state of the Present_Value property from the human operator's viewpoint. The content of this string is a local matter, but it is intended to represent a human-readable description of the ACTIVE state. For example, if the physical input is a switch contact, then the Active_Text property might be assigned a value such as "Fan 1 On".
Inactive_Text	This property, of type CharacterString, characterizes the intended effect of the INACTIVE state of the Present_Value property from the human operator's viewpoint. The content of this string is a local matter, but it is intended to represent a human-readable description of the INACTIVE state. For example, if the physical input is connected to a switch contact, then the Inactive_Text property might be assigned a value such as "Fan 1 Off".
Description	A character string giving more information about the Object associated with the Present_Value property.
Firmware_revision	The firmware revision of the application.
Relinquish_Default	This property is the default value to be used for the Present_Value property when all command priority values in the Priority_Array property have a NULL value.

Appendix D.3. BACnet Vendor ID

BACnet Vendor Name: MSA Safety

BACnet Vendor ID: 37

Appendix D.4. Units

Unit	Variation 1	Variation 2	Type
amperes	amps	A	Electrical
ampere-seconds			Energy
amperes-per-meter			Electrical
amperes-per-square-meter			Electrical
ampere-square-hours			Energy
ampere-square-meters			Electrical
bars			Pressure
becquerels			Other
btus			Energy
btus-per-hour			Power
btus-per-pound			Enthalpy
btus-per-pound-dry-air			Enthalpy
candelas			Light
candelas-per-square-meter			Light
centimeters			Length
centimeters-of-mercury			Pressure
centimeters-of-water			Pressure
cubic-feet			Volume
cubic-feet-per-day			Volumetric Flow
cubic-feet-per-hour			Volumetric Flow
cubic-feet-per-minute			Volumetric Flow
cubic-feet-per-second			Volumetric Flow
cubic-meters			Volume
cubic-meters-per-day			Volumetric Flow
cubic-meters-per-hour			Volumetric Flow
cubic-meters-per-minute			Volumetric Flow
cubic-meters-per-second			Volumetric Flow
currency1			Currency
currency10			Currency
currency2			Currency
currency3			Currency
currency4			Currency
currency5			Currency
currency6			Currency
currency7			Currency
currency8			Currency
currency9			Currency
cycles-per-hour			Frequency
cycles-per-minute			Frequency
days			Time
decibels			Electrical
decibels-a			Other
decibels-millivolt			Electrical
decibels-volt			Electrical
degree-days-Celsius			Temperature
degree-days-Fahrenheit			Temperature
degrees-angular			Other
degrees-Celsius	Deg-C	Deg_C	Temperature
degrees-Celsius-per-hour			Other
degrees-Celsius-per-minute			Other
degrees-Fahrenheit	Deg-F	Deg_F	Temperature
degrees-Fahrenheit-per-hour			Other
degrees-Fahrenheit-per-minute			Other
degrees-Kelvin	Deg-K	Deg_K	Temperature
degrees-Kelvin-per-hour			Temperature
degrees-Kelvin-per-minute			Temperature
degrees-phase			Electrical
delta-degrees-Fahrenheit			Temperature
delta-degrees-Kelvin			Temperature
farads			Electrical
feet			Length
feet-per-minute			Velocity

Unit	Variation 1	Variation 2	Type
feet-per-second			Velocity
foot-candles			Light
grams			Mass
grams-of-water-per-kilogram-dry-air			Humidity
grams-per-cubic-centimeter			Other
grams-per-cubic-meter			Other
grams-per-gram			Other
grams-per-kilogram			Other
grams-per-liter			Other
grams-per-milliliter			Other
grams-per-minute			Mass Flow
grams-per-second			Mass Flow
grams-per-square-meter			Other
gray			Other
hectopascals			Pressure
henrys			Electrical
hertz	Hz		Frequency
horsepower	HP		Power
hours			Time
hundredths-seconds			Time
imperial-gallons			Volume
imperial-gallons-per-minute			Volumetric Flow
inches			Length
inches-of-mercury			Pressure
inches-of-water			Pressure
joule-per-hours			Power
joules			Energy
joule-seconds			Other
joules-per-cubic-meter			Other
joules-per-degree-Kelvin			Entropy
joules-per-kilogram-degree-Kelvin			Entropy
joules-per-kilogram-dry-air			Enthalpy
kilobecquerels			Other
kilo-btus			Energy
kilo-btus-per-hour			Power
kilograms	kg		Mass
kilograms-per-cubic-meter			Other
kilograms-per-hour			Mass Flow
kilograms-per-kilogram			Other
kilograms-per-minute			Mass Flow
kilograms-per-second			Mass Flow
kilohertz	kHz		Frequency
kilohms			Electrical
kilojoules			Energy
kilojoules-per-degree-Kelvin			Entropy
kilojoules-per-kilogram			Energy
kilojoules-per-kilogram-dry-air			Enthalpy
kilometers			Length
kilometers-per-hour			Velocity
kilopascals	Kpa		Pressure
kilovolt-ampere-hours			Energy
kilovolt-ampere-hours-reactive			Energy
kilovolt-amperes	kilovolt-amps	KVA	Electrical
kilovolt-amperes-reactive	KVAR		Electrical
kilovolts			Electrical
kilowatt-hours	kWh		Energy
kilowatt-hours-per-square-foot			Other
kilowatt-hours-per-square-meter			Other
kilowatt-hours-reactive			Energy
kilowatts	kW		Power
liters			Volume
liters-per-hour			Volumetric Flow
liters-per-minute			Volumetric Flow
liters-per-second			Volumetric Flow
lumens			Light
luxes			Light
megabecquerels			Other
mega-btus			Energy
megahertz	MHz		Frequency
megajoules			Energy
megajoules-per-degree-Kelvin			Entropy

Unit	Variation 1	Variation 2	Type
megajoules-per-kilogram-dry-air			Enthalpy
megajoules-per-square-foot			Other
megajoules-per-square-meter			Other
megavolt-ampere-hours			Energy
megavolt-ampere-hours-reactive			Energy
megavolt-amperes	megavolt-amps		Electrical
megavolt-amperes-reactive	MVAR		Electrical
megavolts			Electrical
megawatt-hours	MWh		Energy
megawatt-hours-reactive			Energy
megawatts	MW		Power
megohms			Electrical
meters			Length
meters-per-hour			Velocity
meters-per-minute			Velocity
meters-per-second			Velocity
meters-per-second-per-second			Acceleration
micrograms-per-cubic-meter			Other
micrograms-per-liter			Other
microgray			Other
micrometers			Length
microsiemens			Electrical
microsieverts			Other
microsieverts-per-hour			Other
miles-per-hour			Velocity
milliamperes	milliamps		Electrical
millibars			Pressure
milligrams			Mass
milligrams-per-cubic-meter			Other
milligrams-per-gram			Other
milligrams-per-kilogram			Other
milligrams-per-liter			Other
milligray			Other
milliliters			Volume
milliliters-per-second			Volumetric Flow
millimeters			Length
millimeters-of-mercury			Pressure
millimeters-of-water			Pressure
millimeters-per-minute			Velocity
millimeters-per-second			Velocity
milliohms			Electrical
million-standard-cubic-feet-per-day			Volumetric Flow
million-standard-cubic-feet-per-minute			Volumetric Flow
millirems			Other
millirems-per-hour			Other
milliseconds			Time
millisiemens			Electrical
millisieverts			Other
millivolts			Electrical
milliwatts			Power
minutes			Time
minutes-per-degree-kelvin			Other
mole-percent			Other
months			Time
nanograms-per-cubic-meter			Other
nephelometric-turbidity-unit			Other
newton			Force
newton-meters			Torque
newton-seconds			Other
newtons-per-meter			Other
no-units	No_Units	None	Other
ohm-meters			Electrical
ohm-meter-squared-per-meter			Electrical
ohms			Electrical
parts-per-billion			Other
parts-per-million			Other
pascals			Pressure
pascal-seconds			Other
percent			Other
percent-obscuration-per-foot			Other
percent-obscuration-per-meter			Other

Unit	Variation 1	Variation 2	Type
percent-per-second			Other
percent-relative-humidity	% RH; %RH	Percent RH; Percent RH	Humidity
per-hour			Frequency
per-mille			Other
per-minute			Other
per-second			Other
pH			Other
pounds-force-per-square-inch	PSI	pounds-force-per-sq-inch	Pressure
pounds-mass			Mass
pounds-mass-per-day			Volumetric Flow
pounds-mass-per-hour			Mass Flow
pounds-mass-per-minute			Mass Flow
pounds-mass-per-second			Mass Flow
power-factor	PF		Electrical
psi-per-degree-Fahrenheit			Other
radians			Other
radians-per-second			Other
revolutions-per-minute			Other
seconds	Secs	S	Time
siemens			Electrical
siemens-per-meter			Electrical
sieverts			Other
square-centimeters			Area
square-feet			Area
square-inches			Area
square-meters			Area
square-meters-per-Newton			Other
standard-cubic-feet-per-day			Volumetric Flow
teslas			Electrical
therms			Energy
thousand-cubic-feet-per-day			Volumetric Flow
thousand-standard-cubic-feet-per-day			Volumetric Flow
ton-hours			Energy
tons			Mass
tons-per-hour			Mass Flow
tons-refrigeration			Power
us-gallons	Gallons		Volume
us-gallons-per-hour			Volumetric Flow
us-gallons-per-minute	GPM		Volumetric Flow
volt-ampere-hours			Energy
volt-ampere-hours-reactive			Energy
volt-amperes	Volt-Amps	VA	Electrical
volt-amperes-reactive	VAR		Electrical
volts	voltage		Electrical
volts-per-degree-Kelvin			Electrical
volts-per-meter			Electrical
volt-square-hours			Energy
watt-hours	wH		Energy
watt-hours-per-cubic-meter			Other
watt-hours-reactive			Energy
watts	W		Power
watts-per-meter-per-degree-Kelvin			Other
watts-per-square-foot			Light
watts-per-square-meter			Light
watts-per-square-meter-degree-kelvin			Other
webers			Electrical
weeks			Time
years			Time

Appendix D.5. BACnet Specific Statistics

Stat	Description	Resolution
Link Control	A “who-is” link control message was send or received.	It is normal to receive a few link control messages. If the number is higher than the transmit/receive messages, however, there may be a problem with lost communications.
Unsupported Properties	A request for an unsupported property was received.	This is not an error. BACnet clients often poll all properties of a particular object to determine which properties are supported.
Segmentation Not Supported	Data was requested but the response would have exceeded the maximum size of the APDU and could not be sent using an un-segmented message.	This is not an error – the BACnet client will use a different method to read data from the FieldServer.
Sequence Error	Invoke ID of a reply did not match the Invoke ID of the poll.	You should not see this message. It normally indicates a configuration error.
Write Access Denied	A write to an object was denied.	This typically happens when trying to write to an Input Object that is not Out-Of-Service. It is not possible to write to Input Objects.
Exception Errors	A BACnet Service was denied because it is not supported.	This may be a problem on the Client system. Consult the PIC statement to determine what services are supported.