



**FieldServer**  
**FS-8700-73 BACnet MS/TP - Serial**  
**Driver Manual**  
(Supplement to the FieldServer Instruction Manual)

**APPLICABILITY & EFFECTIVITY**

**Effective for all systems manufactured after November 2015**

Kernel Version: 2.01  
Document Revision: 3

## Contact Information:

Thank you for purchasing the FieldServer.

Please call us for Technical support of the FieldServer product.

### Contact Information:

Sierra Monitor Corporation  
1991 Tarob Court  
Milpitas, CA 95035

Contact number:  
+1 408 262-6611  
+1 800 727-4377

Email: [info@sierramonitor.com](mailto:info@sierramonitor.com)

Website: [www.sierramonitor.com](http://www.sierramonitor.com)

**TABLE OF CONTENTS**

- 1 BACnet MS/TP Description..... 5**
  - 1.1 BTL Mark – BACnet Testing Laboratory..... 5
- 2 Driver Scope of Supply ..... 6**
  - 2.1 Supplied by Sierra Monitor Corporation for this driver ..... 6
  - 2.2 Provided by the Supplier of 3<sup>rd</sup> Party Equipment..... 6
    - 2.2.1 *Hardware* ..... 6
    - 2.2.2 *Optional Items* ..... 6
- 3 Hardware Connections..... 7**
  - 3.1 Hardware connection Tips/Hints..... 8
  - 3.2 Connection Notes ..... 8
- 4 Data Array Parameters ..... 9**
- 5 Configuring the FieldServer as a BACnet MS/TP Client ..... 10**
  - 5.1 Client Side Connection Parameters ..... 10
  - 5.2 Client Side Node Parameters ..... 11
  - 5.3 Client Side Map Descriptor Parameters ..... 11
    - 5.3.1 *FieldServer Specific Map Descriptor Parameters* ..... 11
    - 5.3.2 *Driver Specific Map Descriptor Parameters* ..... 12
    - 5.3.3 *Timing Parameters* ..... 12
    - 5.3.4 *Map Descriptor Example.* ..... 13
- 6 Configuring the FieldServer as a BACnet MS/TP Server..... 14**
  - 6.1 Driver Specific FieldServer Parameters ..... 14
  - 6.2 Server Side Connection Parameters ..... 15
  - 6.3 Server Side Node Parameters..... 16
  - 6.4 Server Side Map Descriptor Parameters..... 16
    - 6.4.1 *FieldServer Specific Map Descriptor Parameters* ..... 16
    - 6.4.2 *Driver Specific Map Descriptor Parameters* ..... 16
    - 6.4.3 *Map Descriptor Example.* ..... 18
- Appendix A. Useful Features ..... 19**
  - Appendix A.1. BACnet object names ..... 19
  - Appendix A.2. Network number..... 19
  - Appendix A.3. Accessing Data from BACnet Properties comprising Arrays of Values..... 20
  - Appendix A.4. FieldServer implementation of BACnet priority Arrays ..... 20
    - Appendix A.4.1. *Accessing Priority Array information* ..... 21
  - Appendix A.5. Relinquishing Control of a Point as a Client ..... 22
  - Appendix A.6. BACnet State Text Preload..... 22
    - Appendix A.6.1. *Method 1 – Using an Offset/User Table:*..... 23
    - Appendix A.6.2. *Method 2 – Using a Single Data Array:*..... 23
    - Appendix A.6.3. *Using Intrinsic Reporting for a Multistate Value* ..... 24
  - Appendix A.7. COV and Intrinsic Reporting ..... 24
    - Appendix A.7.1. *Notes on COV configuration:* ..... 25
    - Appendix A.7.2. *Notes on Intrinsic Reporting configuration:* ..... 25
    - Appendix A.7.3. *Map Descriptor Example - COV*..... 26
    - Appendix A.7.4. *Map Descriptor Example - Enable or Disable COV* ..... 26
    - Appendix A.7.5. *Map Descriptor Example – Intrinsic Reporting*..... 26
    - Appendix A.7.6. *Setting the Priority of Intrinsic Alarming* ..... 26
    - Appendix A.7.7. *Map Descriptor Example – Set up a Permanent Static Recipient List*..... 27
  - Appendix A.8. Factors Determining the Reliability Property ..... 28

Appendix A.9. Specifying ReadPropertyMultiple and WritePropertyMultiple Transactions using Linked Map Descriptors ..... 28

Appendix A.10. Update Property Function ..... 31

*Appendix A.10.1. Supported values:* ..... 31

*Appendix A.10.2. Mode of Operation when used for the Relinquish\_Default property:* ..... 31

Appendix A.11. Disabling selected BACnet services ..... 32

Appendix A.12. Using a .ini file to set the Vendor\_ID and Vendor\_Name of the Device Object ..... 33

Appendix A.13. Specifying the MAC\_Address of the FieldServer using System\_Node\_ID ..... 33

**Appendix B. Troubleshooting** ..... **34**

    Appendix B.1. Debugging a BACnet connection ..... 34

    Appendix B.2. COV Configuration ..... 34

    Appendix B.3. BACnet Specific Statistics ..... 35

    Appendix B.4. BACnet Specific Error Messages ..... 35

    Appendix B.5. BACnet Error Response Decoding ..... 35

*Appendix B.5.1. BACnet Error CLASS* ..... 36

*Appendix B.5.2. BACnet Error CODES for Error Class OBJECT* ..... 37

    Appendix B.6. Rediscovering Offline Devices ..... 38

    Appendix B.7. Timeout Errors ..... 38

    Appendix B.8. PDU limit reached errors ..... 38

**Appendix C. Vendor Information** ..... **39**

    Appendix C.1. McQuay ..... 39

    Appendix C.2. Trane ..... 39

    Appendix C.3. Liebert ..... 39

    Appendix C.4. Automated Logic Corporation ..... 39

    Appendix C.5. Honeywell EBI ..... 39

    Appendix C.6. Using Cimetrics Explorer ..... 39

**Appendix D. Reference** ..... **41**

    Appendix D.1. Object\_Type Legal Values – Abbreviation Descriptions ..... 41

    Appendix D.2. Property Legal Values ..... 41

    Appendix D.3. Units ..... 43

**LIST OF FIGURES**

Figure 1 - Connection Diagram ..... 7

Figure 2 – RS-485 2-Wire Connection Diagram ..... 8

## 1 BACNET MS/TP DESCRIPTION

The BACnet Master-Slave/Token-Passing (MS/TP) driver implements a data link protocol that uses the services of the RS-485 physical layer. See the FieldServer BACnet PIC statement for the level of conformance that this driver implements.

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.

### 1.1 BTL Mark – BACnet Testing Laboratory



BTL is a registered trademark of BTL. BTL is a member company of the BTL Group. BTL is a registered trademark of BTL. BTL is a registered trademark of BTL. BTL is a registered trademark of BTL.

The BTL mark is a symbol that indicates to a consumer that a product has passed a series of rigorous tests conducted by an independent laboratory which verifies that the product correctly implements the BACnet features claimed in the listing. The mark is a symbol of a high-quality BACnet product. For more information about the BACnet Testing Laboratory go to: <http://www.bacnetinternational.net/btl/>.

#### Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	1	
Server	32	

## 2 DRIVER SCOPE OF SUPPLY

### 2.1 Supplied by Sierra Monitor Corporation for this driver

Sierra Monitor Corporation PART #	Description
FS-8917-16	RJ45 to terminal connector cable

### 2.2 Provided by the Supplier of 3<sup>rd</sup> Party Equipment

#### 2.2.1 Hardware

Part #	Description
	BACnet Controller(s) on RS-485 Network

#### 2.2.2 Optional Items

PART #	Vendor/Manufacturer	Description
FS-4301-00	FieldServer	RS-232 to RS-485 converter

### 3 HARDWARE CONNECTIONS

The FieldServer is connected to the BACnet MS/TP as shown in connection drawing.

Configure the BACnet MS/TP according to manufacturer's instructions

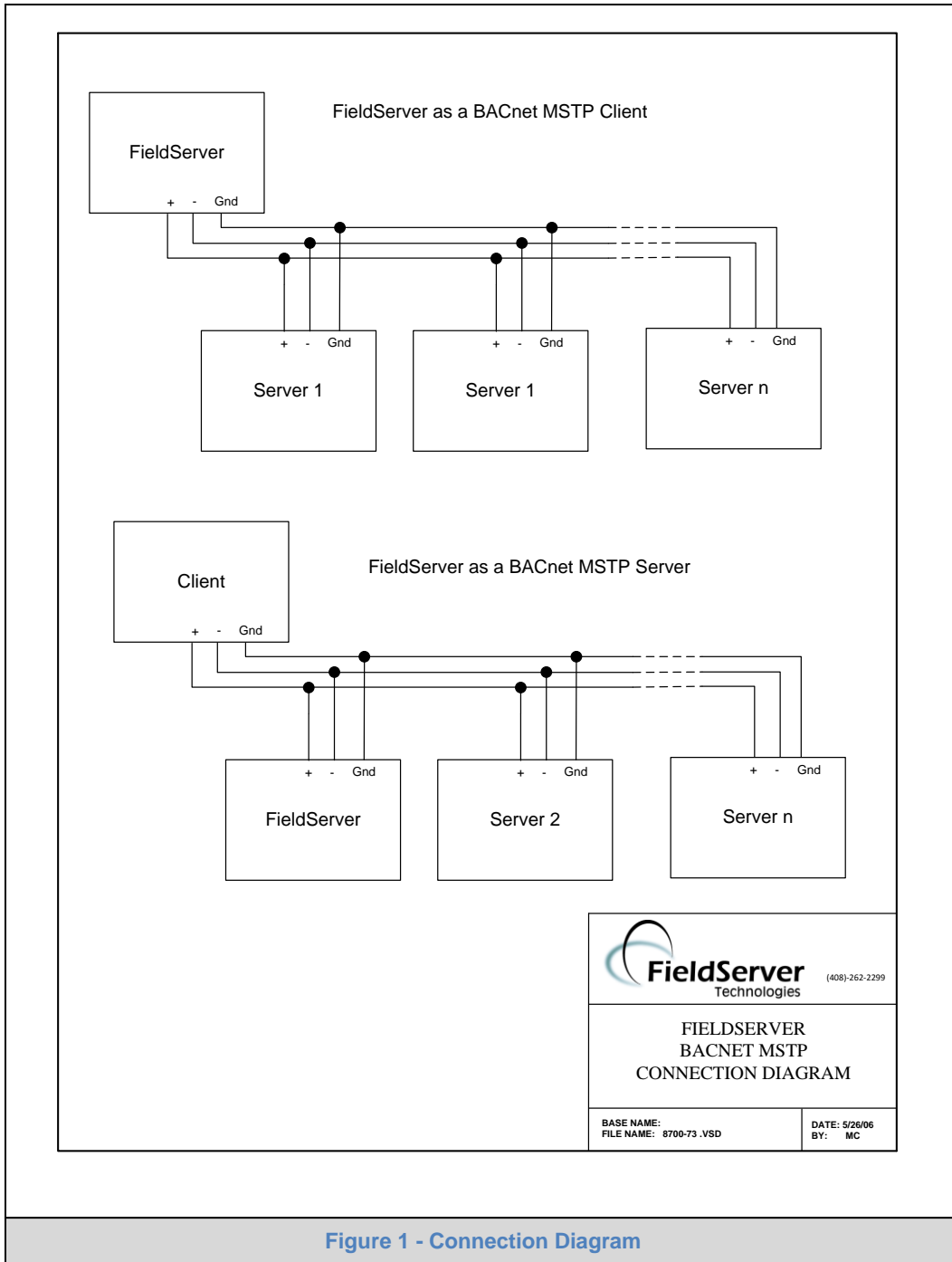
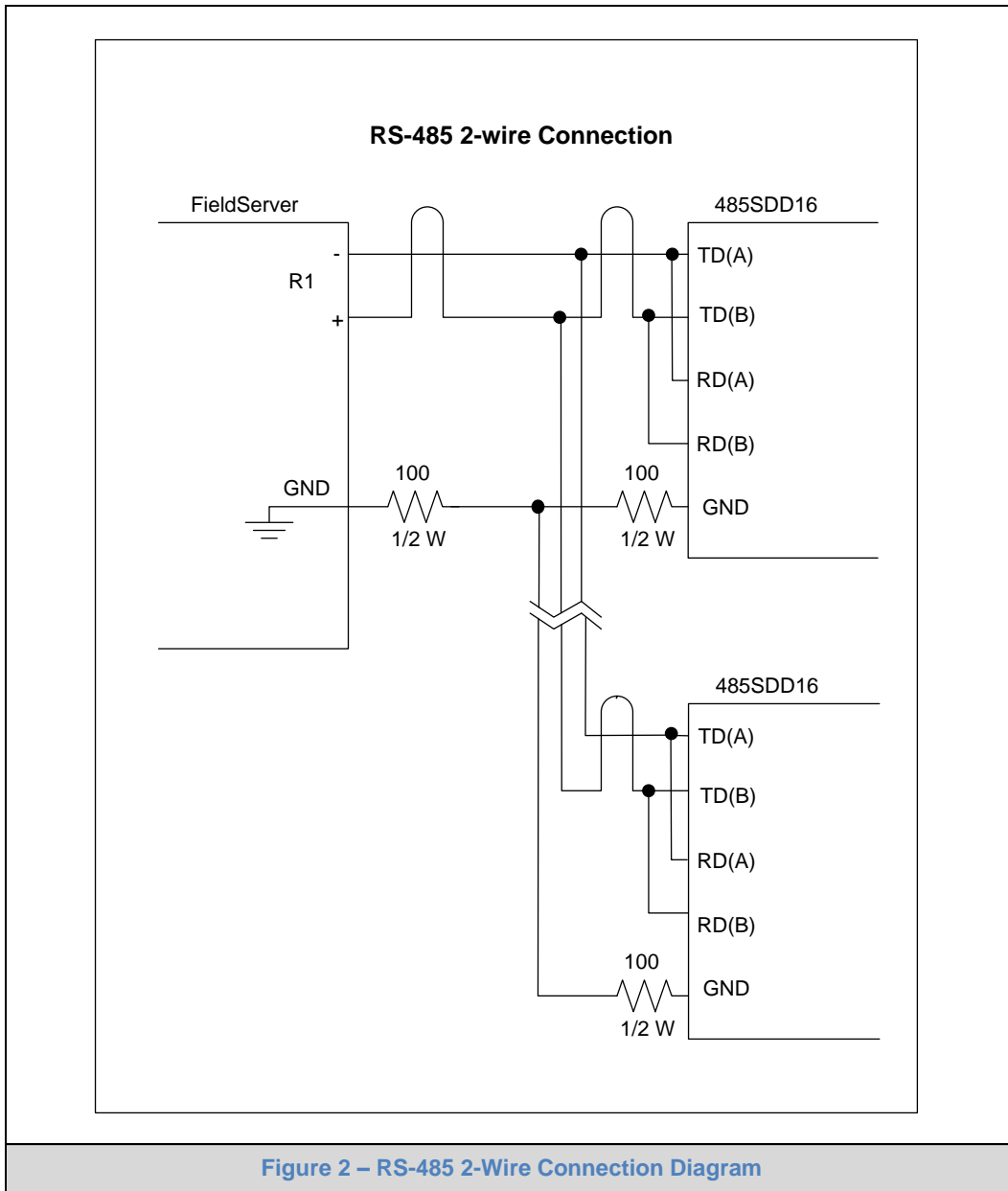


Figure 1 - Connection Diagram

### 3.1 Hardware connection Tips/Hints

Follow standard RS-485 2-wire guidelines for installation. See connection diagram below.



### 3.2 Connection Notes

The QuickServer do not support being configured as a BACnet MS/TP client on one port and a BACnet MS/TP server on another port at the same time. The X30 support this.

In order for this to work the BACnet MS/TP server section must be configured first.



## 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, Uint16, Uint32, Sint16, Sint32, Byte
Data_Array_Length	Number of Data Objects. Must be larger than the data storage area required by the Map Descriptors for data placed in this array.	1-10,000

**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 FIELD SERVER AS A BACNET MS/TP CLIENT

For a detailed discussion on FieldServer configuration, please 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 MS/TP Server.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for BACnet MS/TP 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 that in the tables, \* indicates an optional parameter, with the bold legal value being the default.

### 5.1 Client Side Connection Parameters

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P2 <sup>1</sup> , R1-R2 <sup>2</sup>
Baud	Specify baud rate	<b>9600</b> , 19200, 38400, 76800
Parity*	Specify parity	Odd, Even, <b>None</b>
Data_Bits*	Specify data bits	<b>7</b> , 8
Stop_Bits*	Specify stop bits	<b>1</b> , 2
Protocol	Specify protocol used	BACnet_MS/TP
Poll Delay*	Time between internal polls	0-32000s, <b>1s</b>
Timeout	Application logger timeout	30s
Max_Master*	Specify the highest MAC_Address for Master nodes. When designing a MS/TP network with multiple Master_Node devices, allocate the MAC_Address settings on the different devices starting at 1. This will allow you to specify a small Max_Master parameter and hence will improve overall network performance	<b>1-127</b>
Max_Info_Frames	Specify the maximum number of information frames the connection may send before it must pass the token.	<b>1-65535</b>

#### Example

```
// Client Side Connections

Connections
Port , Baud , Parity , Data_Bits , Stop_Bits , Protocol , Poll_Delay , Timeout
R1 , 38400 , None , 8 , 1 , BACnet_MS/TP , 0.100s , 30s
```

<sup>1</sup> P1 to P2 will require the optional RS-485 converter (Part # 4301-00)

<sup>2</sup> Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

## 5.2 Client Side Node Parameters

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for Node	Up to 31 alphanumeric characters
Node_ID	Specify the BACnet Device object instance.	0 – 4194303
Protocol	Specify protocol used	BACnet_MS/TP
Port	Specify port	P1-P2 <sup>3</sup> , R1-R2 <sup>4</sup>
MAC_Address <sup>5</sup>	Specify the MS/TP MAC address for this connection	0-127 for MS/TP_Master_Mode; 0-254 for MS/TP_Slave_Mode
Network_Number*	Use this parameter if the remote MS/TP server device is on another BACnet network connected via a router.	1, 2, 3.... 16 bit positive numbers, <b>5</b>
Retries*	Number of timeouts before the Node goes offline.	0,1,2,3

Either Node\_ID OR MAC\_Address must be specified. If the Server Node supports Who-Is and I-Am then it is better to use the Node\_ID parameter.

### Example

```
// Client Side Nodes

Nodes
Node_Name , Node_ID , Protocol , Port , Retries
DEV_1 , 1 , BACnet_MS/TP , R1 , 2
```

## 5.3 Client Side Map Descriptor Parameters

### 5.3.1 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. Ensure that unique names are configured for each device. Refer to Appendix A.1.	Up to 37 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

<sup>3</sup> P1 to P2 will require the optional RS-485 converter (Part # 4301-00)

<sup>4</sup> Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

<sup>5</sup> Previously the MAC\_Address was set using System\_Node\_ID: refer to Appendix A.13 for more info

### 5.3.2 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 Section 5.2
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, <b>0</b>
Data_Array_High_Scale*	Scaling max in Data Array	-32767 to 32767, <b>100</b>
Node_Low_Scale*	Scaling zero in Connected Node	-32767 to 32767, <b>0</b>
Node_High_Scale*	Scaling max in Connected Node	-32767 to 32767, <b>100</b>
Length*	Used to create an array of sequential Object_Instances on an Object_Type.	1 to max point count of the FieldServer, <b>1</b>
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, <b>16</b>
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

### 5.3.3 Timing Parameters

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

### 5.3.4 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 FIELD SERVER AS A BACNET MS/TP SERVER

For a detailed discussion on FieldServer configuration, please 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 MS/TP Client.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for BACnet MS/TP 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 that in the tables, \* indicates an optional parameter, with the bold legal value being the default.

### 6.1 Driver Specific FieldServer Parameters

Section Title		
Bridge		
Column Title	Function	Legal Values
Title	FieldServer name	Text
System_Station_Address	The FieldServers MAC address on the MS/TP network	1-254

#### Example

```
// FieldServer Driver specific parameters

Bridge
Title                , System_Station_Address
BacNet MS/TP Server V1.02a  , 11
```

## 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 <sup>6</sup> , R1-R2 <sup>7</sup>
Baud*	Specify baud rate	<b>9600</b> , 19200, 38400, 76800
Parity*	Specify parity	<b>None</b> , Odd, Even
Data_Bits*	Specify data bits	<b>7, 8</b>
Stop_Bits*	Specify stop bits	<b>1, 2</b>
Protocol	Specify protocol used	BACnet_MS/TP
Connection_Type*	Specify if the connection must be in Master_Mode or in Slave_Mode	<b>MS/TP_Master_Mode</b> MS/TP_Slave_Mode
Max_Master*	Specify the highest possible MAC_Address that exists on the MS/TP network. Only applicable for MS/TP_Master Mode connections. (See also section 4.7) <sup>8</sup>	<b>1-127</b>
Max_Info_Frames*	Specify the maximum number of information frames the connection may send before it must pass the token. Only applicable for MS/TP_Master Mode connections. 8	<b>1-65535</b>
MAC_Address <sup>9</sup>	Specify the MS/TP MAC address for this connection	0-127 for MS/TP_Master_Mode; 0-254 for MS/TP_Slave_Mode

### Example

```
// Server Side Connections

Connections
Port , Baud , Parity , Data_Bits , Stop_Bits , Protocol , Timeout , Connection_Type
R1 , 38400 , None , 8 , 1 , BACnet_MS/TP , 30 , MS/TP_Master_Mode
```

If the Server connection needs to operate in Slave Mode then specify the Connection\_Type as MS/TP\_Slave\_Mode

<sup>6</sup> P1 to P2 will require the optional RS-485 converter (Part # 4301-00)

<sup>7</sup> Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

<sup>8</sup> For Server configurations the Max\_Master and Max\_Info\_Frames parameters (or properties) can be overridden from a BACnet Client by using the "Write Property" command. Note that when the FieldServer is restarted, the default values specified in the configuration file will be applied

<sup>9</sup> Previously the MAC\_Address was set using System\_Node\_ID: refer to Appendix A.13 for more info

### 6.3 Server Side Node Parameters<sup>10</sup>

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for Node	Up to 31 alphanumeric characters
Node_ID	BACnet station address of physical Server node	0 – 4194303
Protocol	Specify protocol used	BACnet_MS/TP
Node_Option*	Enable or disable COV for this Node. Refer to Appendix A.7 for further information.	COV_Enable, <b>COV_Disable</b> , -
Model_Name*	Sets the Model Name Property of the Device Object	Up to 32 alphanumeric characters

#### Example

```
// Server Side Nodes

Nodes
Node_Name      , Node_ID  , Protocol      , Model_Name    11
Virtual_DEV_11 , 11      , BACnet_MS/TP , WBA11M. MS/TP
```

### 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 37 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

#### 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 Section 6.3.
Object_Type (Alias = Data_Type)	Data type in Controller	AI, AO, AV, BI, BO, BV, MI, MO, MV, NC. Refer to Appendix A.1.

<sup>10</sup> Only one Server Node may be defined for each connection. For information on support for multiple virtual server nodes on a single connection please contact the Sales Department."

<sup>11</sup> Note that Port is not declared in the Server Side Nodes



Column Title	Function	Legal Values
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, <b>m2</b>
Data_Array_Low_Scale*	Scaling zero in Data Array	-32767 to 32767, <b>0</b>
Data_Array_High_Scale*	Scaling max in Data Array	-32767 to 32767, <b>100</b>
Node_Low_Scale*	Scaling zero in Connected Node	-32767 to 32767, <b>0</b>
Node_High_Scale*	Scaling max in Connected Node	-32767 to 32767, <b>100</b>
Active_Text <sup>12*</sup>	Specify the Active Text property of the Object	Any text string of length up to 40 characters, <b>Active</b>
Inactive_Text <sup>12*</sup>	Specify the Inactive Text property of the Object	Any text string of length up to 40 characters, <b>Inactive</b>
Relinquish_Default <sup>12</sup>	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, <b>1</b>
Notification_Class*	Specify the Mapdescriptor_Name of the Notification_Class Object that manages Intrinsic Reporting for this Map Descriptor. Refer to Appendix A.7.	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.7.	Yes, <b>No</b>
COV_Increment*	For a Server Map Descriptor of type AO or AI, initialize the COV_Increment property. Refer to Appendix A.7.	Any Float value, <b>0</b>
Input_alarm_State*	For binary points the user can specify which of the states (0 or 1) to regard as an alarm state.	<b>0, 1</b>
Description*	Specify the object's description property. E.g. Room Temp – Refer to Appendix A.7.3	Any text string of length up to 40 characters, If not configured, defaults to Object_Name.
Process_ID*	The handle of a process within the recipient device that is to receive the event notification. Refer to Appendix A.7.6.	Any Unsigned integer, -
Notification_Class_MD*	This is the Notification Class map descriptor governing the sending of Event Notifications (Intrinsic Alarming) Refer to Appendix A.7.6	Map Descriptor defined per Appendix A.7

<sup>12</sup> 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].

It is important that unique Map Descriptor names are created for each device to ensure that the Object Names are unique.

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 it's 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, <b>5</b>

**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 please consult FST.

**Note 2:** If an application demands multiple BACnet IP Server nodes and multiple BACnet MS/TP Server Nodes simultaneously, they will at present all appear with the same virtual network number. This creates a potential conformance problem if there is an external BACnet router connecting the BACnet IP and BACnet MS/TP networks to which the FS is also connected, as duplicate virtual networks with the same network number would then exist in the BACnet network, which is not permitted. Please contact FST for assistance.

### 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	Data_Type	Object_ID	Property	Array_Index
CMD_AI_01	DA_AI_01	0	Rdbc	DEV_01	AO	1	Priority_Array	7

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

If no Array\_Index is specified, then the entire property (i.e. the entire BACnet Array) is returned.

The Data\_Index parameter functions as follows for ReadPropertyMultiple or WritePropertyMultiple requests when the Client Map Descriptor length N is greater than 1:

- When an Array\_Index is specified, the ReadPropertyMultiple or WritePropertyMultiple request will iterate through N successive Array\_Index values for the given Object and Property (e.g. Priority Array Index 1 to 16 of Analog Output 1)

- When no Array\_Index is specified, the ReadPropertyMultiple or WritePropertyMultiple request will iterate through N successive Object Instances, reading the specified Property from each successive object (e.g. Present Value of Analog Output 1 to Analog Output 16)

### 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 Data Array where the Priority Array Table will be stored. 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

```
// 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 , Relinquish_default , DA_Pri_Array , DA_Pri_Array_Offset , DA_Pri_In_Use , DA_Pri_In_Use_Offset
CMD_AOP_1 , AO , 1 , Passive , DA_OUT , 0 , N1 11 , 1 , 40.56 , DA_Pri_Array_1 , 0 , DA_Pri_in_use_1 , 0
```

### Appendix A.5. Relinquishing Control of a Point as a Client

It is possible to relinquish control of a point by writing a null to the correct priority level. The following example illustrates how this is done.

Map Descriptor CMD AO is a Read Map Descriptor that will write at priority 7 if a write-through occurs

Map_Descriptors	Map_Descriptor_Name	Data_Type	Function	Scan_Interval	Data_Array_Name	Data_Array_Index	Node_Name	Address	Length	Write_Priority	Service
	CMD AO	, AO	, Rdbc	, 1.0s	, DA AO	, 2	, N1 1	, 1	, 1	, 7	, -
	CMD AO Rel	, AO	, Wrbc	, 1.0s	, DA AO	, 3	, N1 1	, 1	, 1	, 7	, Relinquish

Map Descriptor CMD AO Rel is a Write-on-Change Map Descriptor that will write a NULL at priority 7 (i.e. release Priority Array entry 7) when written to.

Note that the length must be 1.

### Appendix A.6. 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.6.1. Method 1 – Using an Offset/User Table:

```
//set up a look up table

Offset_Table
Offset_Table_Name , Table_String , Table_Index_Value
FIRE_ALARM_TEXT , SYSTEM READY , 1
FIRE_ALARM_TEXT , ALARM , 2
FIRE_ALARM_TEXT , MAINTENANCE , 3
FIRE_ALARM_TEXT , OFF-LINE , 4
FIRE_ALARM_TEXT , IN SERVICE , 5
FIRE_ALARM_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 , Data_Type , Object_ID , Length , State_Text_Array
CMD_MI_01 , DA_MI_01 , 0 , Passive , N1 11 , MI , 1 , 1 , Fire_Alm_Text
```

Appendix A.6.2. 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 MyState4 MyState5 MyState6 , String , 0
```

```
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.6.3. 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_ALARM_TEXT	, SYSTEM READY	, 1	, 0
	FIRE_ALARM_TEXT	, ALARM	, 2	, 1
	FIRE_ALARM_TEXT	, MAINTENANCE	, 3	, 2
	FIRE_ALARM_TEXT	, OFF-LINE	, 4	, 2
	FIRE_ALARM_TEXT	, IN SERVICE	, 5	, 0
	FIRE_ALARM_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.7. 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.

For BACnet/MS/TP, only MS/TP Master Nodes support COV and Intrinsic Reporting

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.7.1. Notes on COV configuration:

COV functionality is not enabled by default for MS/TP. The Node\_Option parameter can be configured to enable or disable COV. An example configuration is presented in Appendix A.7.4. Systems using BACnet/IP or BACnet/Ethernet to BACnet/MS/TP routers may not function reliably if COV functionality is enabled as the router will make the MS/TP device appear as a BACnet/IP or BACnet/Ethernet device, which could lead the BACnet Workstation to expect unrealistic throughput performance when subscribing to multiple points. It is recommended only to enable COV functionality on MS/TP systems if the Client device communicates directly via MS/TP, or if the user knows that the throughput issue will not arise..

COV functionality (SubscribeCOV service) only applies to the Present\_Value and Status\_Flags properties of BACnet Objects - the SubscribeCOVProperty service is 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.

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.

COV Notifications are generated for all data objects for which a remote client has issued a SubscribeCOV-Request. 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.7.2. 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 (specifying which binary value (0 or 1) to regard as the 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 not stored permanently but are kept in volatile memory, and subscriptions must be renewed on system restart.

A permanent static RecipientList may be configured directly on the FieldServer, eliminating the need for a remote workstation to modify the RecipientList in order to receive Event Notifications. Using this option makes the RecipientList read-only, i.e. it can then only be modified via the FieldServer config.csv file. Refer to Appendix A.7.6

### Appendix A.7.3. Map Descriptor Example - 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 , State_Text_Array , Notification_Class , High_Alarm , Low_Alarm , Input_Alarm_State , Confirmed , COV_Increment
SMD_AI_01 , Room Temp , DA_AI_01 , 0 , Passive , Virtual_Dev_11 , AI , 01 , - , - , SMD_NC_01 , 100 , 10 , - , Yes , 1.0
```

### Appendix A.7.4. Map Descriptor Example - Enable or Disable COV

```
Nodes
Node_Name , Node_ID , Protocol , Node_Option
Virtual_Dev_11 , 11 , BACnet_MS/TP , COV_Disable
```

### Appendix A.7.5. Map Descriptor Example – 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.7.6. Setting the Priority of Intrinsic Alarming

Section Title		
Map_Descriptors		
Column Title	Function	Legal Values
Pri_To_Off_Normal*	Priority for off normal events	Refer to table below
Pri_To_Normal*	Priority for normal events	Refer to table below
Pri_To_Fault*	Priority for faults	Refer to table below

A BACnet EventNotification message contains a Priority field indicating the priority of the event being reported. The Priority values to be used are configured via the Priority property of the Notification Class object, using the configuration file parameters Pri\_to\_offnormal, Pri\_to\_normal and Pri\_to\_fault, which determine the priorities to be used respectively for all to-offnormal, to-normal and to-fault transitions reported via the Notification Class object.

Priority Values should be chosen as follows:

Alarm and Event Priority	Network Priority
00-63	Life Safety Message
64-127	Critical Equipment Message
128-191	Urgent Message
192-255	Normal Message

### Appendix A.7.7. Map Descriptor Example – Set up a Permanent Static Recipient List

A permanent static RecipientList may be configured directly on the FieldServer, eliminating the need for a remote workstation to modify the RecipientList in order to receive Event Notifications. Using this option makes the RecipientList read-only, i.e. it can then only be modified via the FieldServer config.csv file.

This is done by adding an Event\_Receiver section to the config.csv. This section must appear after the definition and all other references to the relevant Notification Class object.

Map_Descriptors							
Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_Name	Object_Type	Object_Instance	Ack_Required
BAC_NC_P1	DA_NC_01	, 0	, Passive	Virtual_DEV_11	, NC	, 1	, No

This is the Notification Class map descriptor governing the sending of Event Notifications (Intrinsic Alarming)

Notification Class Objects are configured as Map Descriptors. Each Notification Class Object maintains a RecipientList Property which may contain multiple destinations. Each destination is configured as an Event\_Receiver **after** the relevant Notification Class Object has been configured.

Map_Descriptors									
Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_Name	Object_Type	Object_Instance	Notification_Class	Low_Alarm	High_Alarm
CMD_AI_01	DA_AI_01	, 10	, Passive	Virtual_DEV_11	, AI	, 1	, BAC_NC_P1	, 20	, 30

This is a BACnet Object that will report its alarms via the Notification Class map descriptor BAC\_NC\_P1

Event_Receiver					
Node_Name	Node_ID	Protocol	Process_ID	Notification_Class_MD	Adapter
OWS_01	, 10	, BACnet_MS/TP	, 12345	, BAC_NC_P1	, N1
OWS_02	, 12	, BACnet_MS/TP	, 56789	, BAC_NC_P1	, N1

This section sets up RecipientList entries for the defined Notification Class map descriptor. It must appear at the end of the configuration, after all other references to the Notification Class map descriptor.

## Appendix A.8. Factors Determining the Reliability Property

The Reliability Property of a BACnet object in a Server configuration is determined as follows:

For Multistate Inputs, Outputs and Values, the Reliability property is set to MULTI\_STATE\_FAULT (9) when the Present\_Value of the object corresponds to an entry in the Fault\_Values property of the object, as configured by the State\_Text table in the configuration file.

For all object types, the Reliability property is set to COMMUNICATIONS\_FAILURE (12) when the responsible Server Node is offline.

When neither Condition 1 nor Condition 2 is met, the Reliability property is set to NO\_FAULT\_DETECTED (0)

## Appendix A.9. Specifying ReadPropertyMultiple and WritePropertyMultiple Transactions using 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, a 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	Address	Length	Write_Priority	Linked_Map_Descriptor
CMD RPM 3	, Device	, Max_master	, Passive_Client	, DA RPM	, 0	, N1 1	, 1	, 1	, -	, CMD RPM
CMD RPM 4	, Device	, Max_Info_Frames	, Passive_Client	, DA RPM	, 1	, N1 1	, 1	, 1	, -	, CMD RPM
CMD RPM 2	, AO	, -	, Passive_Client	, DA RPM	, 2	, N1 1	, 8	, 4	, -	, CMD RPM
CMD RPM 1	, AO	, -	, Passive_Client	, DA RPM	, 6	, N1 1	, 4	, 3	, -	, CMD RPM

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

**Notes:**

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 A.10. Update Property Function

The Update\_Property parameter may be configured on a BACnet Server Map Descriptor in order to allow a property other than the Present\_Value to be updated in addition to the Present\_Value when the Data Array value changes.

**Warning:** this is a highly specialized function only to be used under very particular circumstances, by users with a detailed understanding of BACnet and of what is to be achieved.

### Appendix A.10.1. Supported values:

Present\_Value (default operation)

Relinquish\_Default

### Appendix A.10.2. Mode of Operation when used for the Relinquish\_Default property:

When the FieldServer starts up, the standard BACnet operation for AO, AV, BO and BV objects is that the Present\_Value of the object is set to the configured Relinquish\_Default value. This may not be desirable in Gateway situations where the BACnet communications module (i.e. the FieldServer) may restart independently of the controller with the physical outputs.

Instead, it is possible to use the Update\_Property setting to initialize the Relinquish\_Default property using the actual field Present\_Value read from the Client Side device, so that a restart achieves two important outcomes:

The Client side outputs are not modified

The current state of the Client Side output can be read via the Relinquish\_Default property. This is especially useful since the BACnet specification makes no provision for Feedback\_Values on analog objects

Depending on the function of the Client Side Map Descriptor there are slightly differing behaviors.

**RDBC** - If the Client Side Map Descriptor is configured with the RDBC function, then every read operation will cause the Relinquish\_Default property on the BACnet Server side to be updated to match the new Present\_Value.

**ARS** (recommended) - If the Client Side Map Descriptor is configured with the ARS function, then only the startup read operation (which occurs on FieldServer restart, or if the Client Side Node has gone offline and is being recovered) will cause the Relinquish\_Default property on the BACnet Server side to be updated to match the new Present\_Value.

### Restrictions:

This function cannot yet be used in conjunction with Complex data types.

Appendix A.11. Disabling selected BACnet services

Certain BACnet services can be disabled on a BACnet Server Node when specific requirements necessitate this:

COV - certain BACnet MS/TP routers struggle to keep up with the bidirectional traffic that can result from many COVSubscribe requests in quick succession. In this case, better results might be obtained by turning COV off.

WritePropertyMultiple (WPM) - when the user requires writes to the BACnet Server to be acknowledged only once Client Side writes have succeeded, WPM is turned off.

These services can be disabled individually or together by listing them under the optional "Disabled\_Services" parameter of the Node configuration section, separated by a space if more than one function is listed. When a service has been disabled, the BACnet Server Node will reject a corresponding service request with the reason "Unrecognized Service".

The following values may be used under Disabled\_Services:

- WPM (disable support for the WritePropertyMultiple service)
- COV (disable support for the SubscribeCOV service)

**Examples:**

1. Disable WPM only

Nodes			
Node_Name	Node_ID	Protocol	Disabled_Services
NN 01	1	Bacnet_IP	WPM

2. Disable COV only

Nodes			
Node_Name	Node_ID	Protocol	Disabled_Services
NN 02	2	Bacnet_IP	COV

3. Disable WPM and COV

Nodes			
Node_Name	Node_ID	Protocol	Disabled_Services
NN 03	3	Bacnet_IP	WPM COV



Appendix A.12. Using a .ini file to set the Vendor\_ID and Vendor\_Name of the Device Object

The vendor.ini file can be used to change the BACnet Vendor Name and Vendor ID if desired. A file with the following format must be created and downloaded to the FieldServer using the Ruinet “Download Configuration” command.

```
vendor_name = Sierra Monitor Corporation
vendor_id = 37
```

Please note that changing the Vendor\_ID and Vendor\_Name on FieldServer BTL approved products may disqualify them from the BTL approval.

Appendix A.13. Specifying the MAC\_Address of the FieldServer using System\_Node\_ID

The MAC\_Address is used on the Data\_Link (or physical) layer as the source or destination address. See section 9.1.1.2 of the BACnet ANSI/ASHREA standard for more information.

Section Title		
FieldServer		
Column Title	Function	Legal Values
System_Node_ID	Specify the MAC_Address of the FieldServer.	0-254

**Example**

```
// Bridge

Bridge
Title           , System_Node_ID
BACnet_MS/TP   , 11
```

Please note: Specifying the BACnet MAC address using the System\_Node\_ID FieldServer parameter is still supported for backward compatibility with old configuration files, but new configuration files should use the Connection parameters described in section 5.2 and section 6.2.

## 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\_MS/TP connection is configured as a server, it is better to use connection\_type MS/TP\_Slave\_Node. The only time when a server connection must be a Master\_Node is when alarming has been established on the FieldServer.

When a FieldServer/ProtoCessor is configured to support MS/TP virtual server nodes the firmware achieves this by implementing a virtual router and thus a network number is required so that it can expose its nodes to external devices as belonging to a specific network. (The default is network 5 if not specified). This network number is defined on the bridge descriptor as Network\_Number. It must be unique for a site. Refer also to Appendix A.2.

### Appendix B.2. COV Configuration

COV has to be enabled if required for MS/TP. Refer to Appendix A.7 for more information.

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.

**Appendix B.3. 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	Consult the PIC statement to determine what services are supported.

**Appendix B.4. BACnet Specific Error Messages**

Message Description	Cause	Suggested Resolution
Bacnet IP DLL Temporarily out of receive buffers	The FieldServer was flooded with more BACnet IP packets than it could handle. This typically occurs when a workstation discovers a large network resulting in many Who-Is and I-Am broadcasts. The FieldServer will recover from this, but some timeout errors could result.	Take action only if the error is continuous in which case the network load must be analyzed and corrected.

**Appendix B.5. 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.5.1. BACnet Error CLASS

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

**Appendix B.5.2. 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 B.6. Rediscovering Offline Devices

The BACnet driver handles APDU retries internally and uses its own parameters, not the kernel parameters. The kernel timeout and retry values should be left to default.

The following parameters can be configured on the Client Node:

APDU\_Timeout - default value is 10s

APDU\_Retries - default value is 3

The FieldServer will only send requests to the remote device once it has discovered it using the Who-Is / I-Am process.

If a device does not respond and the APDU\_Retries have been used up, the driver will revert to trying to discover the device using Who-Is requests.

Note that there is also a background process of rediscovering devices independently of the polling process. This occurs every 10 minutes.

## Appendix B.7. Timeout Errors

NL TX Abort errors mean that the BACnet network layer was unable to transmit queued messages in time and gave up on them. Outgoing packets are programmed with a 10 second timeout. . If packets have not been transmitted within this 10 second window, they are discarded and an error is recorded.

Timeout errors may be experienced on start-up if the network is not yet fully synchronised or if there are a large number of nodes on the network. The FieldServer can only transmit the I-Ams a few at a time, and only when it receives the MS/TP token.

It might help to give the MS/TP network some time to settle before attempting a discovery.

It may be advisable to discover the Nodes in a number of smaller Node ID ranges (say 10 at a time). The Node\_ID's can be established from the configuration.

## Appendix B.8. PDU limit reached errors

A PDU limit reached error is returned when a device runs out of buffers for storing incoming or outgoing messages (PDU = protocol data unit). A possible scenario is when a large number of server nodes is configured and one or more global Who-Is requests is received. The FieldServer will then need to transmit a response for each one, but if the MS/TP network is busy it might not get enough airtime to send them all out. Another possibility is a buildup of unsent COV or Event Notifications.

## 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.

### 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/MS/TP traffic on its MS/TP port, then the module goes into a terminal mode and MS/TP communications will not be initiated.

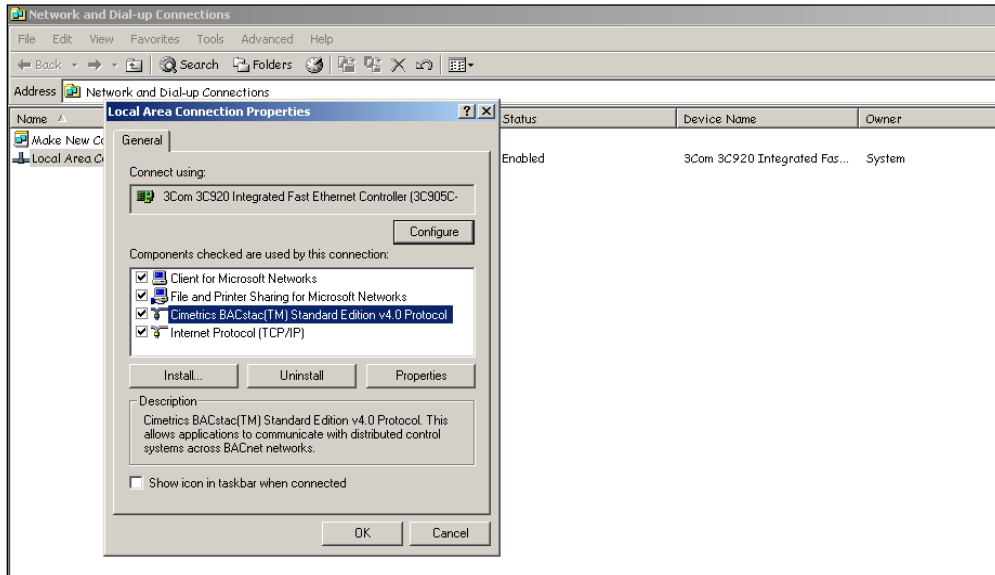
When connecting the FieldServer to an ALC BACnet/MS/TP module, always start the FieldServer first. Wait until the RUN Led is flashing on the FieldServer before powering up the MS/TP module.

### Appendix C.5. Honeywell EBI

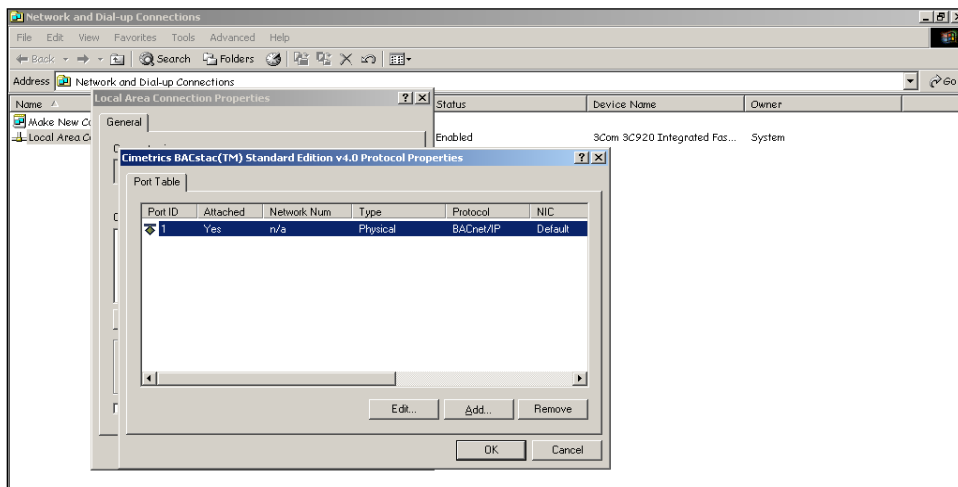
Honeywell EBI cannot process EventNotifications with ACK\_Required set to 1. The ACK\_Required property of Notification Class Map Descriptors configured for use with EBI must therefore be set to 0.

### Appendix C.6. Using Cimetrics Explorer

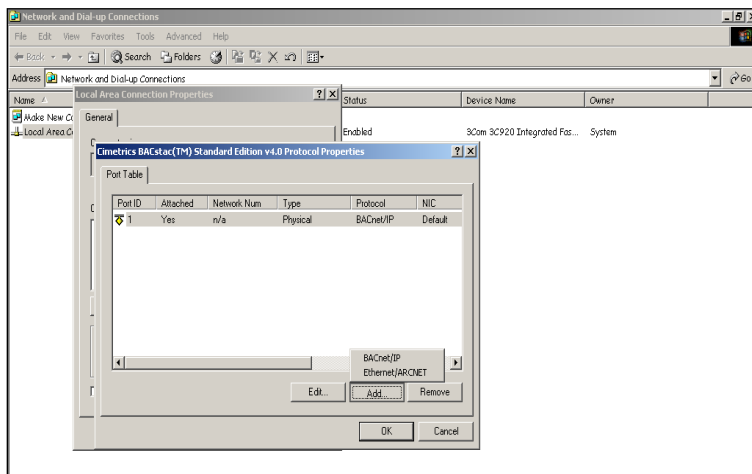
Cimetrics Explorer needs to be configured to use either BACnet Ethernet or BACnet IP. This is not a setting in the actual Cimetrics BACnet Explorer, but is done in the Cimetrics BACstac Protocol that can be found in the Local Area Connection Properties as shown below:



Check the properties of this protocol to see what BACnet protocol is set on the Cimetrics Explorer.



To change the protocol, remove the current one and add a new one as shown below.





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

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_Identifier, 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.

Legal Value	Description
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.
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 writable 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".

Legal Value	Description
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.
Event_Enable	The Event_Enable property is supported, but to save memory it is only instantiated if a point is configured for alarming, i.e. if it is set up with a reference to a Notification Class Map Descriptor.

### Appendix D.3. Units

Unit	Variation 1	Variation 2	Variation 3
<b>Area</b>			
square-centimeters			
square-feet			
square-inches			
square-meters			
<b>Currency</b>			
Currency1			
Currency2			
Currency3			
Currency4			
Currency5			
Currency6			
Currency7			
Currency8			
Currency9			
Currency10			
<b>Electrical</b>			
milliamperes	milliamps		
Amperes	Amps	A	
amperes-per-meter			
amperes-per-square-meter			
ampere-square-meters			
farads			
henrys			
ohms			
ohm-meters			
kilohms			
megohms			
micro_siemens			
siemens			
siemens-per-meter			
TESLAS			
Volts	Voltage		
Voltage			
Millivolts			
kilovolts			
megavolts			
Volt-Amperes	Volt-Amps	VA	
kilovolt-amperes	kilovolt-amps	KVA	
megavolt-amperes	megavolt-amps		

Unit	Variation 1	Variation 2	Variation 3
volt-amperes-reactive	VAR		
kilovolt-amperes-reactive	KVAR		
Megavolt-amperes-reactive	MVAR		
volts-per-degree-kelvin			
volts-per-meter			
degrees-phase			
Power-Factor	PF		
WEBERS			
<b>Energy</b>			
Joules			
kilojoules			
kilojoules-per-kilogram			
megajoules			
watt-hours	Wh		
kilowatt-hours	KWh		
Megawatt-hours	MWh		
BTUs			
Kilo-BTUs			
Mega-BTUs			
Therms			
ton-hours			
kilovolt-amperes--hours			
kilovolt-amperes-reactive-hours			
megavolt-amperes--hours			
megavolt-amperes-reactive-hours			
<b>Enthalpy</b>			
joules-per-kilogram-dry-air			
kilojoules-per-kilogram-dry-air			
megajoules-per-kilogram-dry-air			
btus-per-pound-dry-air			
btus-per-pound			
<b>Entropy</b>			
joules-per-degree-kelvin			
kilojoules-per-degree-kelvin			
megajoules-per-degree-kelvin			
joules-per-kilogram-degree-kelvin			
<b>Force</b>			
newton			
<b>Frequency</b>			
cycles-per-hour			
cycles-per-minute			
Hertz	Hz		
Kilohertz	KHz		
Megahertz	MHz		
Per-hour			
<b>Humidity</b>			
grams-water-per-kg-dry-air			
percent-relative-humidity	% RH; %RH	Percent RH;	PercentRH
<b>Length</b>			
millimeters			
centimeters			
meters			
inches			
feet			
Light			

Unit	Variation 1	Variation 2	Variation 3
candelas			
candelas-per-square-meter			
watts-per-square-foot			
watts-per-square-meter			
lumens			
luxes			
foot-candles			
<b>Mass</b>			
Kilograms	Kg		
Pounds-mass			
Tons			
<b>Mass Flow</b>			
grams-per-second			
grams-per-minute			
kilograms-per-second			
kilograms-per-minute			
kilograms-per-hour			
pounds-mass-per-second			
pounds-mass-per-minute			
pounds-mass-per-hour			
<b>Power</b>			
Milliwatts			
Watts	W		
kilowatts	KW		
megawatts	MW		
BTUs-per-hour			
kilo-btus-per-hour			
Horsepower	HP		
Tons-refrigeration			
<b>Pressure</b>			
Pascals			
hectopascals			
Kilopascals	KPa		
millibars			
Bars			
pounds-force-per-square-inch	PSI	pounds-force-per-sq-inch	
centimeters-of-water			
inches-of-water			
millimeters-of-mercury			
centimeters-of-mercury			
inches-of-mercury			
<b>Temperature</b>			
Degrees-Celsius	Deg-C	Deg_C	
Degrees-Kelvin	Deg-K	Deg_K	
degrees-Kelvin-per-hour			
degrees-Kelvin-per-minute			
Degrees-Fahrenheit	Deg-F	Deg_F	
Degrees-days-Celsius			
Degrees-days-Fahrenheit			
delta-degrees-Fahrenheit			
delta-degrees-Kelvin			
<b>Time</b>			
years			
months			
weeks			

Unit	Variation 1	Variation 2	Variation 3
days			
hours			
minutes			
seconds	Secs	S	
hundredths-seconds			
milliseconds			
<b>Torque</b>			
newton-meters			
<b>Velocity</b>			
millimeters-per-second			
millimeters-per-minute			
meters-per-second			
meters-per-minute			
meters-per-hour			
kilometers-per-hour			
feet-per-second			
feet-per-minute			
miles-per-hour			
<b>Volume</b>			
cubic-feet			
cubic-meters			
imperial-gallons			
liters			
US-gallons	Gallons		
<b>Volumetric Flow</b>			
cubic-feet-per-seconds			
cubic-feet-per-minute			
cubic-feet-per-hour			
cubic-meters-per-seconds			
cubic-meters-per-minute			
cubic-meters-per-hour			
imperial-gallons-per-min			
liters-per-second			
liters-per-minute			
liters-per-hour			
US-Gallons-per-hour			
US-Gallons-per-minute	GPM		
<b>Others</b>			
degrees-angular			
degrees-celsius-per-hour			
degrees-celsius-per-minute			
degrees-fahrenheit-per-hour			
degrees-fahrenheit-per-minute			
joule-seconds			
kilogram-per-cubic-meter			
kilowatt-hour-per-square-meter			
kilowatt-hour-per-square-foot			
megajoules-per-square-meter			
megajoules-per-square-foot			
No-Units	No Units	No_Units	None
newton-seconds			
newtons-per-meter			
parts-per-million			
parts-per-billion			
Percent			

Unit	Variation 1	Variation 2	Variation 3
percent-obscuration-per-foot			
percent-obscuration-per-meter			
percent-per-second			
per-minute			
per-second			
psi-per-degrees-fahrenheit			
radians			
revolutions-per-minute			
square-meters-per-newton			
watts-per-meter-per-degree-kelvin			
watts-per-square-meter-degrees-kelvin			