



FieldServer
FS-8704-06 BACnet/IP
Driver Manual
(Supplement to the FieldServer Instruction Manual)



BTL is a registered trademark of BTL. BTL has not yet been approved for products for use with BACnet/IP. Consideration of this product in the context of BACnet/IP is the responsibility of the user. BTL is a registered trademark of the BACnet Foundation.

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after June 2015

Driver Version: 2.08
Document Revision: 0

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

Website: www.sierramonitor.com

TABLE OF CONTENTS

- 1 BACnet IP 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 3rd Party Equipment..... 6
 - 2.2.1 *Required 3rd Party Hardware*..... 6
- 3 Hardware Connections..... 7**
- 4 Data Array Parameters 8**
- 5 Configuring the FieldServer as a BACnet/IP Client..... 9**
 - 5.1 Client Side Connection Parameters 9
 - 5.2 Client Side Node Parameters 10
 - 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* 11
 - 5.3.3 *Timing Parameters* 12
 - 5.3.4 *Map Descriptor Example* 13
- 6 Configuring the FieldServer as a BACnet/IP Server 14**
 - 6.1 Driver Specific FieldServer Parameters 14
 - 6.2 Server Side Connection Parameters 15
 - 6.3 Server Side Node Parameters..... 15
 - 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* 19
- Appendix A. Useful Features 21**
 - Appendix A.1. Working with BACnet Properties 21
 - Appendix A.1.1. *BACnet object names* 21
 - Appendix A.1.2. *Network number* 23
 - Appendix A.1.3. *Accessing Data from BACnet Properties comprising Arrays of Values* 24
 - Appendix A.1.4. *FieldServer implementation of BACnet priority Arrays*..... 24
 - Appendix A.1.5. *Relinquishing Control of a Point as a Client*..... 25
 - Appendix A.1.6. *BACnet State Text Preload* 26
 - Appendix A.1.7. *Factors Determining the Reliability Property*..... 28
 - Appendix A.1.8. *Update Property Function* 28
 - Appendix A.1.9. *Using a .ini file to set the Vendor_ID and Vendor_Name of the Device Object*..... 30
 - Appendix A.2. Working with BACnet Services 30
 - Appendix A.2.1. *Using BBMD*..... 30
 - Appendix A.2.2. *COV and Intrinsic Reporting* 36
 - Appendix A.2.3. *Specifying ReadPropertyMultiple and WritePropertyMultiple Transactions using Linked Map Descriptors* 41
 - Appendix A.2.4. *Disabling selected BACnet services* 43
 - Appendix A.3. Virtual Router Configuration - connecting a Device to the local BACnet segment..... 44
- Appendix B. Troubleshooting..... 45**
 - Appendix B.1. Debugging a BACnet connection..... 45
 - Appendix B.2. COV Configuration..... 45
 - Appendix B.3. BACnet Specific Statistics..... 46
 - Appendix B.4. BACnet Specific Error Messages 46

Appendix B.5. BACnet Error Response Decoding	46
<i>Appendix B.5.1. BACnet Error CLASS</i>	47
<i>Appendix B.5.2. BACnet Error CODES for Error Class OBJECT</i>	48
Appendix B.6. Rediscovering Offline Devices	49
Appendix C. Vendor Information	50
Appendix C.1. McQuay.....	50
Appendix C.2. Trane.....	50
<i>Appendix C.2.1 Enabling Trane Alarming</i>	50
<i>Appendix C.2.2 Example 1 – Analog input</i>	51
<i>Appendix C.2.3 Example 2 – Binary Input</i>	51
Appendix C.3. Liebert	52
Appendix C.4. Honeywell EBI	52
Appendix C.5. Using Cimetrics Explorer	52
Appendix C.6. Trending using OniconTrend Log Objects	53
<i>Appendix C.6.1 Operating Statistics</i>	54
<i>Appendix C.6.2 Date and Time Synchronization</i>	55
Appendix D. Reference	56
Appendix D.1. FieldServer Vendor ID	56
Appendix D.2. Object_Type Legal Values – Abbreviation Descriptions	56
Appendix D.3. Note on Configuring Binary Outputs	56
Appendix D.4. Property Legal Values	57
Appendix D.5. Units.....	58

LIST OF FIGURES

Figure 1 - Generic Connection Diagram	7
---	---

1 BACNET IP DESCRIPTION

The BACnet/IP driver allows the FieldServer to transfer data to and from devices over Ethernet using BACnet/IP protocol. 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. See Appendix D.1 for abbreviation list.

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 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



The BTL Mark on ProtoNode RER is a symbol that indicates 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.

Go to <http://www.BACnetInternational.net/btl/> for more information about the BACnet Testing Laboratory. Click here for [BACnet PIC Statement](#)

2 DRIVER SCOPE OF SUPPLY

2.1 Supplied by Sierra Monitor Corporation for this driver

Sierra Monitor Corporation PART #	Description
FS-8915-10	UTP cable (7 foot) for Ethernet connection ¹

2.2 Provided by the Supplier of 3rd Party Equipment

2.2.1 Required 3rd Party Hardware

Part #	Description
	Ethernet 10/100 BaseT hub ²

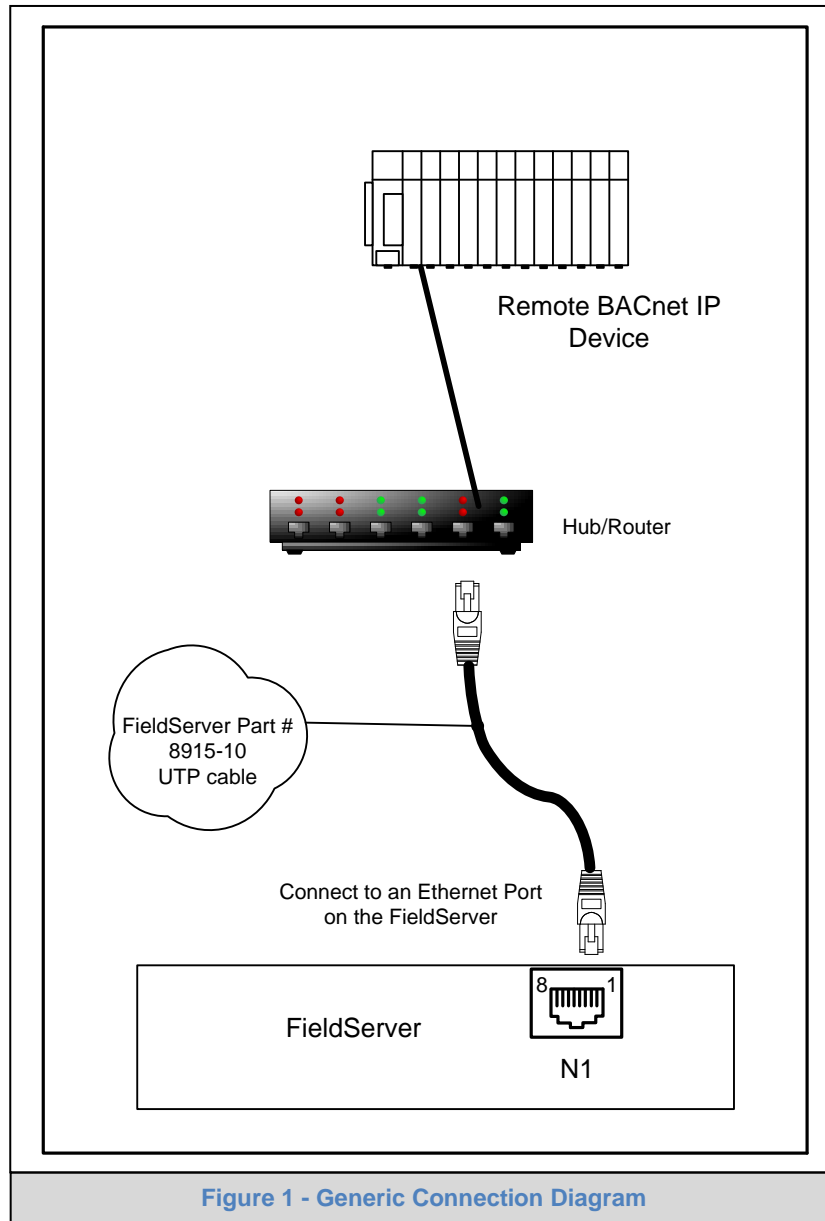
¹ This cable is necessary for connection to the driver. It is shipped with the FieldServer and not separately with the driver.

² Not all FieldServer models support 100BaseT. Consult the appropriate instruction manual for details of the Ethernet speed supported by specific hardware.

3 HARDWARE CONNECTIONS

It is possible to connect a BACnet/IP device using the N1 or the N2³ network ports. These ports need to be configured for BACnet/IP in the configuration file.

Configure the Controller according to manufacturer's instructions.



³ Not all ports shown are necessarily available on 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, Uint16, Sint16, Uint32, Sint32, Byte.
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.	1-10, 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 FIELDSEVER AS A BACNET/IP 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/IP Server.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for BACnet/IP 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-P8, R1-R2 ⁴
Protocol	Specify protocol used	
Poll_Delay*	Time between internal polls	0-32000s; 0.25s
IP_Port*	Specify the UDP port that will be used to communicate with other BACnet Client devices.	Any legal IP port value (1 - 65535); 47808
Connection_Type*	Specify if this Connection should act as a BBMD Client on the network. Refer to Appendix A.2.1 for more information.	BBMD, -

Example

```
// Client Side Connections

Connections
Adapter , Protocol , Poll_Delay
N1 , Bacnet_IP , 0.1s
```

⁴ 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	BACnet device identifier of physical Server Node	0 - 4194303
Protocol	Specify protocol used	BACnet_IP
Adapter	Specify port Adapter used	N1, N2Error! Bookmark not defined.
Retries*	Number of timeouts before the Node goes offline.	0,1,2,3
APDU_Timeout*	The time in milliseconds between retransmissions of an APDU requiring acknowledgement for which no acknowledgment has been received.	10000ms
APDU_Retries*	The maximum number of times that an APDU shall be retransmitted.	3
MAC_Address*	Configuring the remote MAC address directly replaces the default operation (dynamic device binding) using Who-Is broadcasts. This is useful in situations where either the network or the target device do not support dynamic device binding (e.g. when talking to MS/TP slave devices). <u>Note:</u> the Device Instance / Node_ID should not be specified when the MAC Address is configured directly.	The remote device IP Address and Port, e.g. 192.168.1.174:47808 or a 6-byte octet string in hexadecimal format, e.g. c0:a8:1:ae:ba:c0

Note:

Permitted Values (APDU_Timeout and APDU_Retries):

The total delay in seconds before giving up on a transmission (APDU_Timeout/1000 * (APDU_Retries/1000 + 1)) should not exceed 65 seconds.

Example

```
// Client Side Nodes

Nodes
Node_Name , Node_ID, , Protocol , Adapter , Retries
DEV_01 , 1 , Bacnet_IP , N1 , 2
```

5.3 Client Side Map Descriptor Parameters

5.3.1 FieldServer Specific Map Descriptor Parameters

Section Title		
Map Descriptors		
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.1.1.	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. Refer to the FieldServer Configuration manual for more information.	Rdbc, Wrbc, Wrbx, Arcs

5.3.2 Driver Specific Map Descriptor Parameters

Section Title		
Map Descriptors		
Column Title	Function	Legal Values
Node_Name	Name of remote Server Node.	One of the Node_Names specified in Section Error! Reference source not found.
Object_Type (Alias = Data_Type)	Type of object. Refer to Appendix A.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 0
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	0 to max number of array items in the BACnet Property, -

	Appendix A.1.3	
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.2.3	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.2.3
COV Specific		
If the remote server supports COV-B it is possible to configure the FieldServer BACnet Client to subscribe to COV updates. This can be done instead of or in addition to periodic reads. The advantage of using COV is that changed values will be reported within a much shorter time, especially if a large number of points is being monitored. However, not all BACnet devices support COV, so exercise care when using this function.		
Service	BACnet Point service	COVSubscribe
Confirmed	Used to select Confirmed or Unconfirmed COV Notifications	Yes, No
Function	Set to ARS or AWS to subscribe once on startup, or to RDBC or WRBC to re-subscribe after each Scan_Interval	ARS, AWS, RDBC, WRBC
COV_Lifetime	Specify the COV subscription lifetime in seconds. Use a large value to get the benefit of decreased network communications.	Value in seconds
Scan_Interval	If using RDBC in order to re-subscribe periodically, set the Scan_Interval to the desired re-subscription interval. This should be significantly shorter than COV_Lifetime.	Value in seconds

5.3.3 Timing Parameters

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

5.3.4 Map Descriptor Example

```
// Client Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Object_Type , Object_Instance , Property , Scan_Interval
CMD_AI_01 , DA_AI_01 , 0 , Rdbc , DEV_01 , AI , 1 , Present_Value , 20.000s
CMD_AI_02 , DA_AI_01 , 1 , Rdbc , DEV_01 , AI , 2 , Present_Value , 20.000s
CMD_AI_03 , DA_AI_01 , 2 , Rdbc , DEV_01 , AI , 3 , Present_Value , 20.000s
CMD_AO_01 , DA_AO_01 , 0 , Rdbc , DEV_01 , AO , 1 , Present_Value , 30.000s
CMD_AO_02 , DA_AO_01 , 1 , Rdbc , DEV_01 , AO , 2 , Present_Value , 30.000s
CMD_AO_03 , DA_AO_01 , 2 , Rdbc , DEV_01 , AO , 3 , Present_Value , 30.000s

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Object_Type , Object_Instance , Property , Scan_Interval
CMD_DI_01 , DA_DI_01 , 0 , Rdbc , DEV_01 , BI , 1 , Present_Value , 15.000s
CMD_DI_02 , DA_DI_01 , 1 , Rdbc , DEV_01 , BI , 2 , Present_Value , 15.000s
CMD_DI_03 , DA_DI_01 , 2 , Rdbc , DEV_01 , BI , 3 , Present_Value , 15.000s
CMD_DO_01 , DA_DO_01 , 0 , Rdbc , DEV_01 , BO , 1 , Present_Value , 30.000s
CMD_DO_02 , DA_DO_01 , 1 , Rdbc , DEV_01 , BO , 2 , Present_Value , 30.000s
CMD_DO_03 , DA_DO_01 , 2 , Rdbc , DEV_01 , BO , 3 , Present_Value , 30.000s
```

5.3.4.1 COV Specific

In this example, Map Descriptor `CMD_AI_01_SUB` creates and periodically renews the COV subscription. Map Descriptor `CMD_AI_01_RD` performs periodic reads of the same object and will also be updated by COV Notifications received.

```
Map Descriptor
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Data_Type , Object_ID , Service , COV_Lifetime , Confirmed , Scan_Interval
CMD_AI_01_SUB , DA_AI , 0 , RDBC , BCU_01 , AI , 0 , COVSubscribe , 600 , Yes , 300
CMD_AI_01_RD , DA_AI , 1 , RDBC , BCU_01 , AI , 0 , - , - , - , 30
```

6 CONFIGURING THE FIELDSEVER AS A BACNET/IP 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” files provided with the FieldServer).

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

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for BACnet/IP 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 Client’s 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
Network_Number*	Specify a unique network number if there are multiple virtual Server Nodes. Refer to Appendix A.1.2	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
Adapter	Adapter Name	N1, N2 ⁵
Protocol	Specify protocol used	BACnet_IP
Connection_Type*	Specify if this Connection should act as a BBMD Server on the network. Refer to Appendix A.2.1 for more information.	BBMD, -
IP_Port*	Specify the UDP port that will be used to communicate with other BACnet Client devices.	Any legal IP port value (1 - 65535); 47808
Net_Addr_Len*	Set the address length of the virtual network used to connect multiple virtual BACnet devices to the external network. By default this address is 6 bytes long; older FieldServer versions used 2 or 4 bytes, and if a legacy system depends on a length different to 6, this can be configured here. Note that since the virtual network address corresponds directly to the Node ID, the Node ID values should not exceed the maximum value that can be encoded in the number of address bytes specified.	1-6, -

Example

```
// Server Side Connections

Connections
Adapter      , Protocol      , Connection_Type
N1           , Bacnet_IP   , BBMD
```

6.3 Server Side Node Parameters

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_IP
Node_Option*	Enable or disable COV for this Node. Refer to Appendix A.2.1.3 for further information.	COV_Enable , COV_Disable, -
Node_Type*	Specific to Trane applications. The Node type can be defined. Refer to Appendix A. Appendix C.2	BCU, Summit Workstation, -
Disabled_Services*	Certain BACnet Services can be disabled on a BACnet Server Node when specific requirements necessitate this. Refer to Appendix A.2.4 for more information.	-, WPM, COV

⁵ Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

Model_Name*	Sets the Model Name Property of the Device Object	Up to 32 alphanumeric characters
Network_Location*	Specifies on which network the virtual node is located	-, Local Segment. Refer to Appendix A.3

Example

```
// Server Side Nodes

Nodes
Node_Name      , Node_ID  , Protocol  , Node_Option  , Model_Name
Virtual_DEV_11 , 11      , Bacnet_IP , COV_Enable  , WBA11M. MSTP BACnet-IP
```

6.4 Server Side Map Descriptor Parameters

6.4.1 FieldServer Specific Map Descriptor Parameters

Section Title		
Map Descriptors		
Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor. This is used for Object_Name Property. Refer to Appendix A.1.1.	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

Section Title		
Map Descriptors		
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.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 A.Appendix D.5, m2

⁶ Note that adapter is not declared under Server Side Nodes.

Section Title		
Map Descriptors		
Column Title	Function	Legal Values
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 ^{7*}	Specify the Active Text property of the Object	Any text string of length up to 40 characters, Active
Inactive_Text ^{7*}	Specify the Inactive Text property of the Object	Any text string of length up to 40 characters, Inactive
Relinquish_Default ⁷	Specify the value to be returned as Present_Value on startup or when control is relinquished. Must be specified for outputs. Required for AO, AV, BO, BV, MO and MV Data Types only. Refer to 0	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.2.1.3.	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.2.1.3.	Yes, No
COV_Increment*	For a Server Map Descriptor of type AO or AI, initialize the COV_Increment property. Refer to Appendix A.2.1.3.	Any Float value, 0
Input_Alarm_State*	This parameter is required when a BI, BO or BV Map Descriptor is configured for alarms, i.e. when a Notification_Class is specified for the Map Descriptor. It defines the value (0 or 1) that is to be treated as the alarm (i.e. off-normal) value.	0, 1 (there is no default value)

⁷ See the BACnet DFS to determine if a particular object supports this property

Section Title		
Map Descriptors		
Column Title	Function	Legal Values
Description*	Specify the object's description property.	Any text string of length up to 40 characters, e.g. Room Temp (refer to Appendix A.2.2.4), or a Data Array name and start offset that holds the text string in the following format: <Data Array name.offset> e.g. <DA_AI_01_Desc.0> If not configured, defaults to Object_Name.
Update_Property*	The Update_Property Parameter may be configured on a BACnet Server Map Descriptor to allow a property other than the Present_Value to be updated <i>in addition</i> 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. Refer to 0	Present Value, Relinquish_Default
Notification_Class_MD*	This is the Notification Class map descriptor governing the sending of Event Notifications (Intrinsic Alarming) Refer to Appendix A.2.2.7	Map Descriptor defined per Appendix A.2.1.3
Profile_Parameter*	Specify the profile parameter that can be updated by remote BACnet client using regular BACnet objects. The FieldServer will keep these parameters in non-volatile memory.	Any profile parameter.
Min_Pres_Value*	Specify the "Minimum Present Value" property for an AO Object. This setting can also be used on AV, even though the AV object itself does not support the property.	Any floating point value is legal. If not specified, default to +/- Infinity
Max_Pres_Value*	Specify the "Maximum Present Value" property for an AO Object. This setting can also be used on AV, even though the AV object itself does not support the property.	Any floating point value is legal. If not specified, default to +/- Infinity

Note: In this implementation the Min_Pres_Value & Max_Pres_Value properties can be read but not written via BACnet.

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
```

```
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Object_Type , Object_Instance , Relinquish_Default , Min_Pres_Value , Max_Pres_Value
SMD_11_AO_03 , DA_AO_01 , 0 , Passive , Virtual_Dev_11 , AO , 03 , 0 , 10 , 20
```

6.4.3.1 Example for using Profile Parameters

The following example config is used to update BACnet network number, Device ID and the UDP port number dynamically. The following defines could be in another profile config.

```
// Default Defines

Defines
Name , Value
BN_Network_Nr , 50
BN_Node_Offset , 500
BN_UDP_Port_Nr , 6666
```

```
// Common Information

FieldServer
Title           , Network_Number
Test BAC_IP Client V1.15I , $BN_Network_Nr
```

```
// Data Arrays

Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_MV_01       , UInt32      , 3
```

```
// Server Side Connections

Connections
Adapter       , Protocol   , IP_Port
N1            , Bacnet_IP  , $BN_UDP_Port_Nr
```

```
// Server Side Nodes

Nodes
Node_Name    , Node_ID      , Protocol
Dev_01       , $BN_Node_Offset , Bacnet_IP
```

```
// Server Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Object_Type , Object_Instance , Profile_Parameter
SMD_MV_01           , DA_MV_01        , 0                  , Passive , Dev_01    , MV          , 01              , BN_Network_Nr
SMD_MV_02           , DA_MV_01        , 1                  , Passive , Dev_01    , MV          , 02              , BN_Node_Offset
SMD_MV_03           , DA_MV_01        , 2                  , Passive , Dev_01    , MV          , 03              , BN_UDP_Port_Nr
```

Appendix A. USEFUL FEATURES

Appendix A.1. Working with BACnet Properties

Appendix A.1.1. BACnet object names

When an external BACnet Client builds a list of Object Names, the BACnet Server Map Descriptor names and Object Types (AI, AO, BI, BO, MI, MO, etc.) 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 Object Type is AI, the 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]. The maximum length of a point Object Name is the same as the maximum Map_Descriptor_Name length. Refer to Section **Error! Reference source not found.**

The Device Object Name is set from either the Node Name or the Model Name, and the maximum length is determined by the maximum number of characters defined in the legal values column. Refer to Section 6.3

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

Appendix A.1.1.1. Using Tags to create unique Device and Object names

The <device_id> tag can be used to tag the actual device object instance to the end of the specified Object_Name.

Nodes			
Node_Name	, Node_ID	, Protocol	, Node_Option
Dev_IP_<device_id>	, 11	, Bacnet_IP	, COV_Enable

In the example above the FieldServer will appear as Dev_IP_11 when it is discovered on BACnet. The Map_Descriptor reference to the BACnet node must also include the same <device_id> tag. If the Node_ID is now changed using the dip-switches, then the Device Object_Name will automatically change with the same value at the end.

Map_Descriptors							
Map_Descriptor_Name	, Data_Array_Name	, Data_Array_Offset	, Function	, Node_Name	, Data_Type	, Object_ID	, Relinquish_Default
SMD_DI_01	, DA_DI_01	, 0	, Server	, Virtual_BCU_<device_id>	, DI	, 1	, -

Appendix A.1.2. 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 1: 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 MSTP 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 MSTP 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.1.3. 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	Object_Type	Object_Instance	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.1.4. 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.1.4.1. 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 , Object_Type , Object_Instance , Function , Data_Array_Name , Data_Array_Offset , 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 1 , 1 , 40.56 , DA_Pri_Array_1 , 0 , DA_Pri_in_use_1 , 0
```

Appendix A.1.5. 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_Des	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.1.6. 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.1.6.1. 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
```

```
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  , Object_Type  , Object_Instance  , Length  , State_Text_Array
CMD_MI_01            , DA_MI_01        , 0                  , Passive  , N1 11     , MI          , 1                , 1       , Fire_Arm_Text
```

Appendix A.1.6.2. 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	Object_Type	Object_Instance	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.1.6.3. 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 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.1.7. 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.1.8. 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.1.8.1. Supported values

- Present_Value (default operation)
- Relinquish_Default

Appendix A.1.8.2. 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.1.8.3. Update Property Configuration Example:

Map_Descriptors									
Map_Descriptor_Name	Function	Data_Array_Name	Data_Array_Index	Node_Name	Data_Type	Address	Length	Relinquish_Default	Update_Property
AOP1	Passive	DA AOP	0	NN 01	AO	1	1	10	Relinquish_Default

Appendix A.1.9. 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.2. Working with BACnet Services

Appendix A.2.1. Using BBMD

A BBMD (BACnet Broadcast Management Device) is used to allow devices on different subnets to communicate to one another. BACnet/IP requires that a BBMD be defined on every subnet. The FieldServer can act as a BBMD for the subnet that it resides on. Setting the Connection_Type to BBMD will enable this functionality on the FieldServer.

Notes:

- BBMD operation is not required if there is already another BBMD on the subnet – there can only be one BBMD per subnet.

Appendix A.2.1.1. Setting up a Broadcast Distribution Table (BDT)

The FieldServer supports two methods of updating its internal broadcast distribution table.

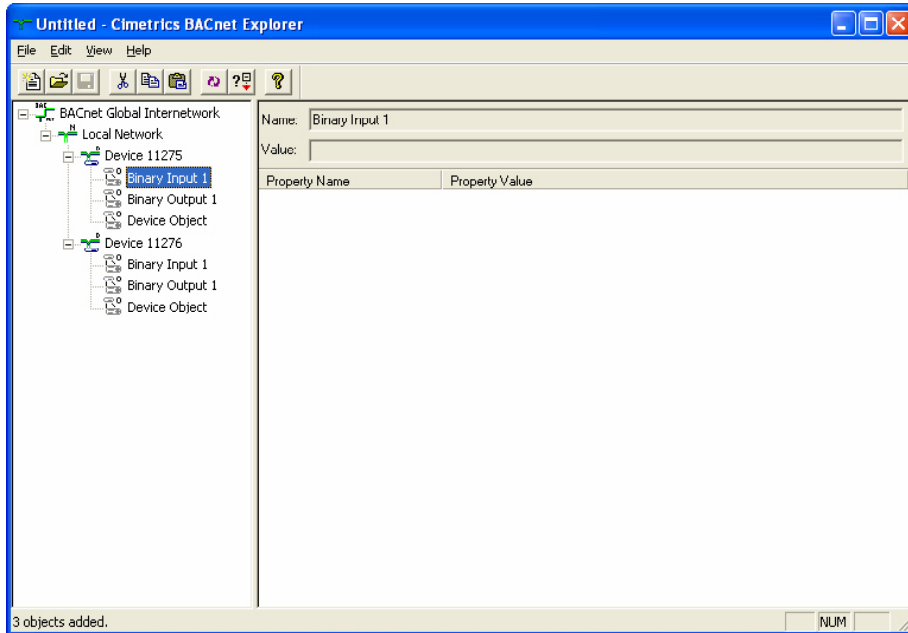
- Trane’s Tracer Summit IP Validation utility can be used directly to upload and download a Broadcast Distribution Table.
- The Broadcast Distribution Table can be configured using the bdt.ini file. The following file must be created and downloaded to the FieldServer using the Ruinet “Download Configuration” command.

```
// Bdt.ini
// The format of this table must be :
//
// BBMD IP_Address , BBMD port , BBMD subnet Mask
24.90.48.179 , 47808 , 255.255.255.255
64.80.115.156 , 47808 , 255.255.255.255
```

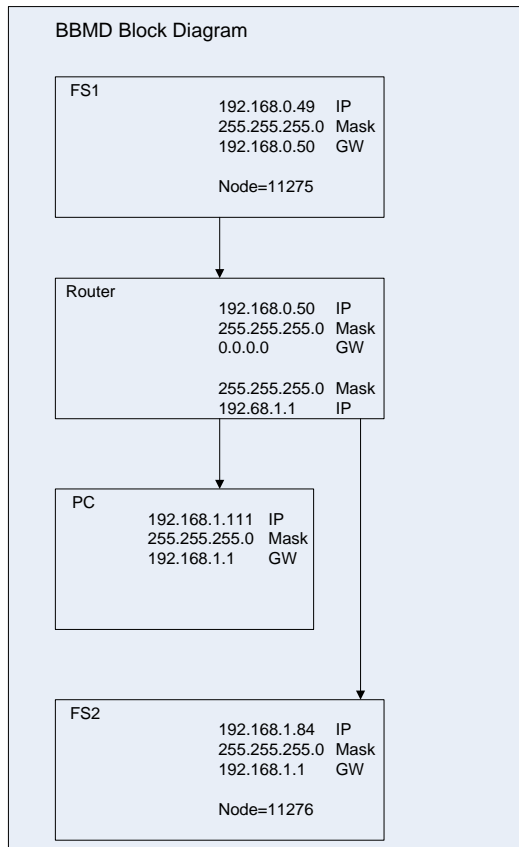
Note: BACnet will create an empty broadcast distribution table should a bdt.ini file not exist. BACnet/IP network tools can then directly update the broadcast distribution table over the network and a bdt.ini file will be created automatically.

Appendix A.2.1.2. BBMD Configuration Example

Using the following configuration, a PC using BACnet explorer is able to browse and display the contents of two FieldServers. The following image shows the BACnet explorer screen.



The following diagram shows the layout of the connections between the FieldServers and the PC via a router.



```
// Bdt.ini
// The format of this table must be :
//
// BBMD IP_Address , BBMD port , BBMD subnet Mask
192.168.1.84 , 47808 , 255.255.255.255
192.168.0.49 , 47808 , 255.255.255.255
```

Configuration of the FieldServers is shown in the following examples.

FieldServer # 1

```
// Common Information

Bridge
Title           , System_Address
Server-1 Pumping Package #1 , 1
```

```
// Data Arrays

Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AI           , UINT16      , 200
DA_AO           , UINT16      , 200
DA_DI           , Bit         , 200
DA_DO           , Bit         , 200
```

```
Connections
Adapter         , Protocol   , Connection_Type , IP_Port
N1              , Bacnet_IP  , BBMD            , 47808
```

```
Nodes
Node_Name      , Node_ID    , Protocol
Server-1       , 11275     , Bacnet_IP
```

```
// Server Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Object_Type , Object_Instance , Units , Relinquish_Default
SCHWRremoteCall    , DA_DO           , 0                 , Passive , Server-1 , BO          , 1              , No-Units , 0
SCHWCommonAlarm    , DA_DI           , 0                 , Passive , Server-1 , BI          , 1              , No-Units , -
```

FieldServer # 2

```
// Common Information
Bridge
Title           , System_Address
Server-1 Pumping Package #2 , 1
```

```
// Data Arrays
//
Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
DA_AI           , UINT16      , 200
DA_AO           , UINT16      , 200
DA_DI           , Bit          , 200
DA_DO           , Bit          , 200
```

```
Connections
Adapter        , Protocol   , Connection_Type , IP_Port
N1             , Bacnet_IP  , BBMD           , 47808
```

```
Nodes
Node_Name     , Node_ID    , Protocol
Server-1     , 11276     , Bacnet_IP
```

```
// Server Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Object_Type , Object_Instance , Units , Relinquish_Default
HWRemoteCall       , DA_DO           , 0                 , Passive , Server-1 , BO          , 1              , No-Units , 0
HWCommonAlarm      , DA_DI           , 0                 , Passive , Server-1 , BI          , 1              , No-Units , -
```

Appendix A.2.1.3. Communicating through a NAT router

The FieldServer BBMD can be configured to operate through a NAT router, thus making the BBMD as well as all BACnet Devices configured directly on the same FieldServer accessible to remote BACnet clients via the Internet, e.g. using Foreign Device registration.

Note:

The FieldServer does not yet implement a BACnet router, and hence only BACnet devices and objects configured directly on the FieldServer will be accessible via the NAT router. Other devices may become visible because of broadcast forwarding, but the FieldServer will not route incoming requests or responses to any other devices on the same local network.

Configuration:

The Global_IP_Address and Global_IP_Port settings must be added to the Connection configuration item as shown below. The Global_IP_Address and Global_IP_Port must be set to the externally visible IP Address and UDP Port of the NAT router that will be forwarded to the local BACnet IP Address and Port of the FieldServer.

Connections				
Adapter	Protocol	Connection_Type	Global_IP_Address	Global_IP_Port
N1	Bacnet_IP	BBMD	105.236.196.77	47808

Appendix A.2.2. 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. The value specified for the COV_Increment is not affected by Map Descriptor scaling parameters, ie. If the COV_Increment=1 the value in the Data Array must change by at least 1 for a COV notification to be sent, even if the Data Array Values are scaled.
- 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 and for all subscribed client workstations to be notified of alarm acknowledgements (using EventNotifications of type ACK_NOTIFICATION).

Appendix A.2.2.1. 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.2.2.3
- 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.2.2.2. 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 volatile, 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.2.2.7
- The “Event Enable” property is required for any object supporting Intrinsic Alarming. This property is supported, but to save memory 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 A.2.2.3. Node Configuration Example - Enable or Disable COV

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

Appendix A.2.2.4. Map Descriptor Example - COV

```
// Notification Class Object

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Object_Type , Object_Instance , 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 , Object_Type , Object_Instance , 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.2.2.5. 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.2.2.6. 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.2.2.7. 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	Address
OWS_01	10	BACnet_MSTP	12345	BAC_NC_P1	N1
OWS_02	12	BACnet_MSTP	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.2.3. 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	0	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.2.4. 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.3. Virtual Router Configuration - connecting a Device to the local BACnet segment

When the FieldServer is configured with multiple BACnet Devices (Nodes) it creates an internal, virtual BACnet segment and acts as a virtual router in order to make all the Devices individually addressable via a single external MAC Address.

However, some 3rd party BACnet utilities depend on the presence of a BACnet Device directly on the local BACnet segment (i.e. not on the internal, virtual segment) to be able to discover the FieldServer. A local BACnet device can be created especially for this purpose as shown in the example below.

Nodes			
Node_Name	Node_ID	Protocol	Network_Location
Virtual_Dev_11	11	Bacnet_IP	Local_Segment
Virtual_Dev_12	12	Bacnet_IP	-

In the example, Virtual_Dev_11 will appear on the local BACnet segment, whereas Virtual_Dev_12 will appear on a remote BACnet segment identified by the Network_Number assigned to the FieldServer. Only one Node may be configured to appear on the Local Segment.

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.
- Some of the BACnet IP features result in the creation of files (priarray.ini; desc.ini; alarms.ini) on the FieldServer. Sometimes updates of firmware can result in these files becoming outdated. Deleting these files will restore configuration defaults and may assist with configuration errors.
- 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 using the FieldServer as a BACnet Server, ensure the FieldServer's Subnet Mask matches the Subnet Mask of the BACnet Client. Otherwise, communications are very slow and eventually stop altogether.
- If the FieldServer gateway is unable to discover the target device, then check that the Subnet masks are the same with the IP addresses with in the subnet address range, or simply set the FieldServer Subnet mask to a larger class (e.g: 255.255.0.0) to see if this helps. If it does help, check with the responsible System Administrator to ensure that a wider broadcast won't generate other issues

Appendix B.2. COV Configuration

- 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	This may be a problem on the Client system. Consult the PIC statement for supported services.

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

Error Description	Code	Suggested Resolution
Other	0	
Authentication failed	1	Verify username and Password on device
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 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.2.1 Enabling Trane Alarming

The Driver will automatically support Trane Alarming if the Node_Type parameter as defined in Section 6.3 is set to BCU or Summit Workstation.

The following Trane Specific Map Descriptor Parameters also need to be defined.

Section Title		
Map Descriptors		
Column Title	Function	Legal Values
Event_Receiver_Name	The name of the BACnet Node to which alarms must be sent. This Node must be configured under the Nodes section of the configuration file.	One of the Node_Names specified in Section 6.3
Low_Alarm	For AI, AO and AV object types this sets the lower alarm limit.	Any floating point value
High_Alarm	For AI, AO and AV object types this sets the upper alarm limit.	Any floating point value
Event_Class*	A numerical field used to group Events into classes	Any integer value, 0
Ack_Required*	Specifies whether or not an Alarm messages must be acknowledged by the workstation	Yes, No
Alarm_Deadband*	For AI, AO, and AV types, this is the amount by which the present value must be greater than the Low_Alarm limit or less than the High_Alarm limit for the object to return to the Normal state and send a to-normal event.	Any floating point value, 0.0
Input_Alarm_State*	For BI, BO and BV types this defines the value (0 or 1) that is to be treated as the alarm (i.e. off-normal) value.	0, 1

Appendix C.2.2 Example 1 – Analog input

Map_Descriptors													
Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_Name	Data_Type	Object_ID	Length	Event_Receiver_Name	Low_Alarm	High_Alarm	Event_Class	Ack_Required	Alarm_Deadband
ABB_vfd1_Output Frequency	ai1vfd1	, 1	, Passive	, Bac11	, AI	, 1	, 1	, ADD_WS	, 20	, 30	, 5	, Yes	, 2
ABB_vfd1_Current	ai1vfd1	, 2	, Passive	, Bac11	, AI	, 2	, 1	, ADD_WS	, 0.5	, 1.5	, 5	, Yes	, 0.2

Appendix C.2.3 Example 2 – Binary Input

Map_Descriptors											
Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_Name	Data_Type	Object_ID	Length	Event_Receiver_Name	Input_Alarm_State	Event_Class	Ack_Required
ABB_VFd1_RO_1 Status	bi1vfd1	, 0	, Passive	, Bac11	, BI	, 1	, 1	, ADD_WS	, 1	, 5	, Yes
ABB_VFd1_Run Status	bi1vfd1	, 1	, Passive	, Bac11	, BI	, 2	, 1	, ADD_WS	, 1	, 5	, Yes

Appendix C.3. Liebert

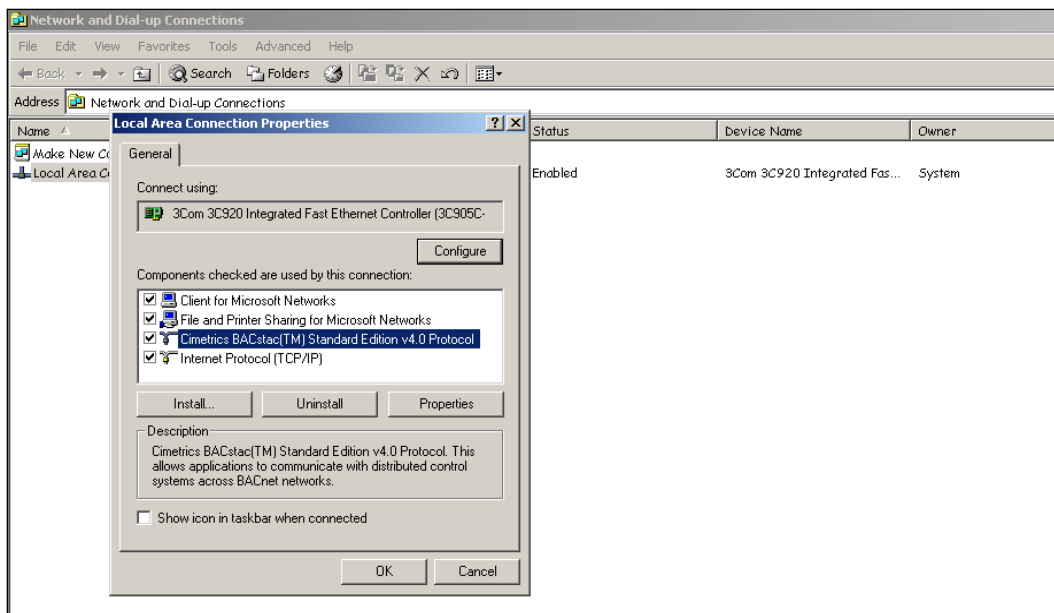
Polling BACnet addresses that are not configured for Liebert systems may cause the connection to fail in older versions of Liebert. Please contact your Liebert supplier for more information.

Appendix C.4. Honeywell EBI

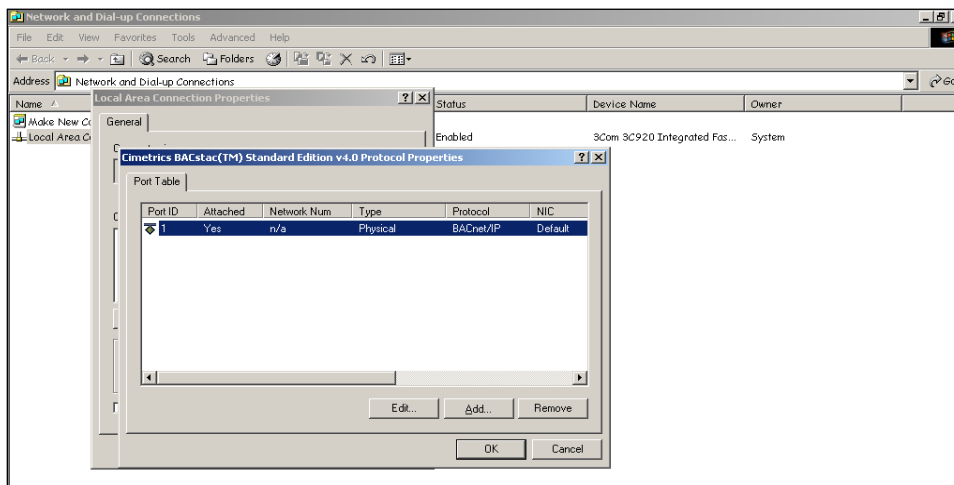
Honeywell EBI cannot process EventNotifications with ACK_Required set to 1. The ACK_Required property of Notification Class Map Descriptors configured for use with EBI must therefore be set to 0.

Appendix C.5. Using Cimetrics Explorer

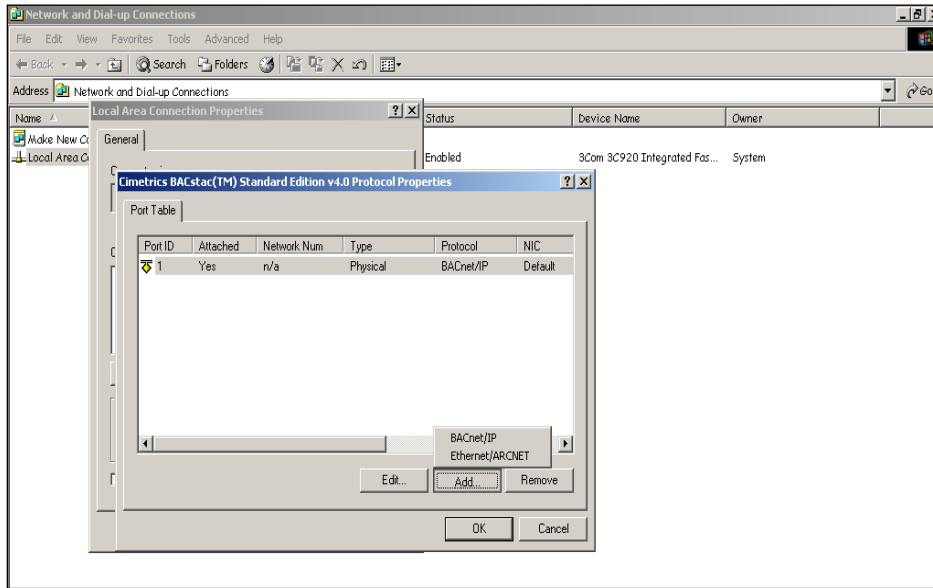
Cimetrics Explorer needs to be configured to use either BACnet Ethernet or BACnet IP. This is not a setting in the actual Cimetrics BACnet Explorer, but is done in the Cimetrics BACstac Protocol that can be found in the Local Area Connection Properties as shown below:



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 C.6. Trending using OniconTrend Log Objects

The FieldServer BACnet driver has been updated to support trending using Trend Log objects. A Map Descriptor is required for each Trend log to be added to the device. The following parameters are specific to this application.

Section Title		
Map Descriptors		
Column Title	Function	Legal Values
Log_Data_Array	The Data Array containing the data which the Trend Log will log	One of the Data Arrays specified in Section 4.
Log_Data_Array_Index	The position in the Data Array of the data value to be logged	0 to (Data_Array_Length -1), -
Log_Buffer_Size*	The maximum number of records that will be kept in the log buffer. This should be limited due to the size constraints of the disk.	0 - 65535 log entries
Log_Interval*	How often a record will be stored, specified in seconds.	4 - 4294967295
Log_Enable*	This setting initializes the Enable property of the Trend Log at start-up. Logging only takes place while Enable is True ("Yes"). Note that this property is writable from BACnet, the user has the option to set it to "No" in the configuration file and allow the BACnet Client to set the Enable property as required.	Yes, No

Notes:

- The Client will need to read the log every "Log Interval x Log Buffer Size" to prevent losing data; e.g. 500 records x 4 seconds / record = 2000 seconds = 33.3 Minutes.
- In this version the Stop_When_Full property is not writable or configurable, and is set to False.

```
// Server Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name, Data_Array_Name, Data_Array_Index, Function, Node_Name, Data_Type, Address, Log_Data_Array, Log_Data_Array_Index, Log_Buffer_Size, Log_Interval, Log_Enable
Trend Log 1      , DA_TREND      , 0      , Passive , BTU METER, Trend_Log, 0      , DA_AI_01      , 0      , 500      , 4      , Yes
Trend Log 2      , DA_TREND      , 1      , Passive , BTU METER, Trend_Log, 1      , DA_AI_01      , 1      , 500      , 4      , Yes
Trend Log 3      , DA_TREND      , 2      , Passive , BTU METER, Trend_Log, 2      , DA_AI_01      , 2      , 500      , 4      , Yes
Trend Log 4      , DA_TREND      , 3      , Passive , BTU METER, Trend_Log, 3      , DA_AI_01      , 3      , 500      , 4      , Yes
```

Appendix C.6.1 Operating Statistics

The following stats were added for diagnosing the operation of the Trend Log, they can be viewed with Ruinet.

BACnet connection stats	
Stat	Description
Pending Trend Log Writes	The number of writes waiting to be written to the disk.

BACnet Trend Log Map Descriptor Stats:	
Stat	Description
Total Record Count	Number of entries recorded
Record Count	Current number of entries within the log, this will stop at the value specified by the configuration
Status Records Added	Number of status events
Data Records Added	Number of data events
Time Change Records Added	Number of time change events

Digi nI9210 LED
The blue LED will flash periodically when an event log is stored to the disk, this light should not be on all the time, otherwise we might be building up a back-log of outstanding writes.

Appendix C.6.2 Date and Time Synchronization

The QuickServer system time is not set on startup but depends on an external time synch to initialize its clock. This can be done via BACnet time synch and should be done whenever comms to the device have been down since this could indicate a restart. Since the time values are only correct after this synch it is best for the BACnet Client to perform a time synch before enabling trend logging.

If the device is configured to start logging by itself from restart the initial timestamps will start at Jan 1, 1970, and a time change record will be created on time synch.

Also note that the latest Ruinet will automatically set the system time if it connects to a QuickServer with an uninitialized system time.

Appendix D. REFERENCE

Appendix D.1. FieldServer Vendor ID

BACnet Vendor Name: Sierra Monitor Corporation
BACnet Vendor ID: 37

Appendix D.2. Object_Type Legal Values – Abbreviation Descriptions

Abbreviation	Associated BACnet Number	Description
AI	0	ANALOG_INPUT
AO	1	ANALOG_OUTPUT
AV	2	ANALOG_VALUE
BI	3	BINARY_INPUT
BO	4	BINARY_OUTPUT
BV	5	BINARY_VALUE
MI	13	MULTI_STATE_INPUT
MO	14	MULTI_STATE_OUTPUT
MV	19	MULTI_STATE_VALUE
NC	15	NOTIFICATION_CLASS_OBJECT
DEVICE	8	DEVICE

Appendix D.3. Note on Configuring Binary Outputs

The BACnet specification defines the behavior of Binary Outputs such that the Present_Value property is treated as a set-point and is only expected to change as a result of BACnet write requests from upstream, and not as a result of values read from downstream. This may cause confusion in situations where a user wants to map BACnet Binary Output objects to corresponding Binary Output points on a downstream device, since the Present_Value property will not be updated to reflect the value of the downstream point. Instead, the Feedback_Value property may be monitored in order to know the state of the downstream point.

For alarms (EventNotifications) this implies that BACnet BO points will trigger a COMMAND_FAILURE alarm, which will trigger if the Feedback_Value (read from the downstream) differs from the Present_Value.

In order to be able to use the Present_Value property to read and write to a downstream binary output object, configure a Binary Value object on the FieldServer.

Appendix D.4. 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.
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.

Legal Value	Description
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.1.6).
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".
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.
Min_Pres_Value	Specify the "Minimum Present Value" property.
Max_Pres_Value	Specify the "Maximum Present Value" property

Appendix D.5. Units

Unit	Variation 1	Variation 2	Variation 3
Acceleration			
meters-per-second-per-second			
Area			
square-meters			
square-centimeters			
square-feet			
square-inches			
Currency			
currency1			

currency2			
currency3			
currency4			
currency5			
currency6			
currency7			
currency8			
currency9			
currency10			
Electrical			
milliamperes	milliamps		
amperes	Amps	A	
amperes-per-meter			
amperes-per-square-meter			
ampere-square-meters			
decibels			
decibels-millivolt			
decibels-volt			
farads			
henrys			
ohms			
ohm-meters			
milliohms			
kilohms			
megohms			
microsiemens			
millisiemens			
siemens			
siemens-per-meter			
teslas			
volts	Voltage		
millivolts			
kilovolts			
megavolts			
volt-amperes	Volt-Amps	VA	
kilovolt-amperes	kilovolt-amps	KVA	
megavolt-amperes	megavolt-amps		
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			
Energy			
joules			
kilojoules			
kilojoules-per-kilogram			
megajoules			
watt-hours	Wh		
kilowatt-hours	KWh		
megawatt-hours	MWh		
watt-hours-reactive			
kilowatt-hours-reactive			
megawatt-hours-reactive			
btus			
kilo-btus			
mega-btus			
therms			
ton-hours			
Enthalpy			
joules-per-kilogram-dry-air			
kilojoules-per-kilogram-dry-air			
megajoules-per-kilogram-dry-air			
btus-per-pound-dry-air			
btus-per-pound			
Entropy			
joules-per-degree-Kelvin			
kilojoules-per-degree-Kelvin			
megajoules-per-degree-Kelvin			
joules-per-kilogram-degree-Kelvin			
Force			
newton			
Frequency			
cycles-per-hour			
cycles-per-minute			
hertz	Hz		
kilohertz	KHz		
megahertz	MHz		
per-hour			
Humidity			
grams-water-per-kg-dry-air			
percent-relative-humidity	% RH; %RH	Percent RH;	PercentRH
Length			
micrometers			
millimeters			

centimeters			
kilometers			
meters			
inches			
feet			
Light			
candelas			
candelas-per-square-meter			
watts-per-square-foot			
watts-per-square-meter			
lumens			
luxes			
foot-candles			
Mass			
milligrams			
grams			
kilograms	Kg		
pounds-mass			
tons			
Mass Flow			
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			
tons-per-hour			
Power			
milliwatts			
watts	W		
kilowatts	KW		
megawatts	MW		
btus-per-hour			
kilo-btus-per-hour			
horsepower	HP		
tons-refrigeration			
Pressure			
pascals			
hectopascals			
kilopascals	KPa		
millibars			
bars			

pounds-force-per-square-inch	PSI	pounds-force-per-sq-inch	
millimeters-of-water			
centimeters-of-water			
inches-of-water			
millimeters-of-mercury			
centimeters-of-mercury			
inches-of-mercury			
Temperature			
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			
Time			
years			
months			
weeks			
days			
hours			
minutes			
seconds	Secs	S	
hundredths-seconds			
milliseconds			
Torque			
newton-meters			
Velocity			
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			
Volume			
cubic-feet			
cubic-meters			
imperial-gallons			
milliliters			

liters			
us-gallons	Gallons		
Volumetric Flow			
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-minute			
milliliters-per-second			
liters-per-second			
liters-per-minute			
liters-per-hour			
us-gallons-per-minute	GPM		
us-gallons-per-hour			
Other			
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-hours-per-square-meter			
kilowatt-hours-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			
percent-obscuration-per-foot			
percent-obscuration-per-meter			
percent-per-second			
per-minute			
per-second			
psi-per-degree-Fahrenheit			
radians			
radians-per-second			
revolutions-per-minute			
square-meters-per-Newton			

watts-per-meter-per-degree-Kelvin			
watts-per-square-meter-degrees-kelvin			
per-mille			
grams-per-gram			
kilograms-per-kilogram			
grams-per-kilogram			
milligrams-per-gram			
milligrams-per-kilogram			
grams-per-milliliter			
grams-per-liter			
milligrams-per-liter			
micrograms-per-liter			
grams-per-cubic-meter			
milligrams-per-cubic-meter			
micograms-per-cubic-meter			
nanograms-per-cubic-meter			
grams-per-cubic-centimeter			
becquerels			
kilobecquerels			
megabecquerels			
gray			
milligray			
microgray			
sieverts			
millisieverts			
microsieverts			
microsieverts-per-hour			
decibels-a			
nephelometric-turbidity-unit			
pH			
grams-per-square-meter			
minutes-per-degree-kelvin			