



FieldServer
FS-8704-02 BACnet Ethernet
Driver Manual
(Supplement to the FieldServer Instruction Manual)



BTL is a registered trademark of BTL. BTL is not responsible for any products that do not conform to BTL standards. The use of BTL products is the responsibility of the user. BTL is a registered trademark of the BTL Group.

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after March 2017.

Driver Revision: 2.01
Document Revision: 4.A

Technical Support

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

Sierra Monitor Corporation
1991 Tarob Court
Milpitas, CA 95035

Website: www.sierramonitor.com

U.S. Support Information:

+1 408 262-6611

+1 800 727-4377

Email: support@sierramonitor.com

EMEA Support Information:

+44 2033 1813 41

Email: support.emea@sierramonitor.com

TABLE OF CONTENTS

- 1 BACnet Ethernet Description 5**
 - 1.1 BTL Mark – BACnet Testing Laboratory..... 5
- 2 Driver Scope of Supply 5**
 - 2.1 Supplied by Sierra Monitor Corporation 5
 - 2.2 Provided by the Supplier of 3rd Party Equipment..... 5
- 3 Hardware Connections..... 6**
- 4 Data Array Parameters 7**
- 5 Configuring the FieldServer as a BACnet Ethernet Client 8**
 - 5.1 Client Side Connection Parameters 8
 - 5.2 Client Side Node Parameters 8
 - 5.3 Client Side Map Descriptor Parameters 9
 - 5.3.1 FieldServer Related Map Descriptor Parameters 9
 - 5.3.2 Driver Related Map Descriptor Parameters 10
 - 5.3.3 Timing Parameters 11
 - 5.3.4 Map Descriptor Examples 11
- 6 Configuring the FieldServer as a BACnet Ethernet Server 12**
 - 6.1 Server Side Connection Parameters 12
 - 6.2 Server Side Node Parameters..... 13
 - 6.3 Server Side Map Descriptor Parameters..... 13
 - 6.3.1 FieldServer Related Map Descriptor Parameters 13
 - 6.3.2 Driver Specific Map Descriptor Parameters 14
 - 6.3.3 Map Descriptor Example 15
- Appendix A. Useful Features 16**
 - Appendix A.1. BACnet Object Names..... 16
 - Appendix A.2. Virtual Network Number..... 16
 - Appendix A.3. Accessing Data from BACnet Properties Comprising Arrays of Values 17
 - Appendix A.4. FieldServer Implementation of BACnet Priority Arrays..... 17
 - Appendix A.4.1. Accessing Priority Array information 18
 - Appendix A.5. Relinquishing Control of a Point as a Client 19
 - Appendix A.6. BACnet State Text Preload..... 19
 - Appendix A.6.1. Method 1 – Using an Offset/User Table..... 19
 - Appendix A.6.2. Method 2 – Using a Single Data Array..... 20
 - Appendix A.6.3. Using Intrinsic Reporting for a Multistate Value 20
 - Appendix A.7. COV and Intrinsic Reporting 21
 - Appendix A.7.1. COV Configuration Tips 21
 - Appendix A.7.2. Intrinsic Reporting Configuration Tips 22
 - Appendix A.7.3. Map Descriptor Example - Enable or Disable COV 22
 - Appendix A.7.4. Map Descriptor Example - COV 22
 - Appendix A.7.5. Map Descriptor Example – Intrinsic Reporting..... 22
 - Appendix A.7.6. Setting the Priority of Intrinsic Alarming 23
 - Appendix A.7.7. Map Descriptor Example – Set up a Permanent Static Recipient List..... 23
 - Appendix A.8. Factors Determining the Reliability Property 24
 - Appendix A.9. Specifying ReadPropertyMultiple and WritePropertyMultiple Transactions Using Linked Map Descriptors 24
 - Appendix A.10. Update Property Function 26
 - Appendix A.10.1. Supported Values..... 26
 - Appendix A.10.2. Mode of Operation when Used for the Relinquish_Default Property 26
 - Appendix A.10.3. Update Property Configuration Example..... 26
 - Appendix A.11. Disabling Selected BACnet Services..... 27
 - Appendix A.12. Using a .ini File to Set the Vendor_ID and Vendor_Name of the Device Object 27
 - Appendix A.13. Writing to Custom Properties on Remote BACnet Devices..... 28

Appendix B. Troubleshooting.....	29
Appendix B.1. Debugging a BACnet connection.....	29
Appendix B.2. COV Configuration.....	29
Appendix B.3. BACnet Specific Statistics.....	29
Appendix B.4. BACnet Specific Error Messages	30
Appendix B.5. BACnet Error Response Decoding	30
Appendix C. Vendor Information.....	31
Appendix C.1. McQuay.....	31
Appendix C.2. Trane.....	31
Appendix C.2.1. Enabling Trane Alarming	31
Appendix C.2.2. Example 1 – Analog input	32
Appendix C.2.3. Example 2 – Binary Input.....	32
Appendix C.3. Liebert.....	32
Appendix C.4. Honeywell EBI	32
Appendix C.5. Using Cimetrics Explorer	33
Appendix D. Reference.....	34
Appendix D.1. FieldServer Vendor ID	34
Appendix D.2. Object_Type Legal Values – Abbreviation Descriptions	34
Appendix D.3. Property Legal Values	35
Appendix D.4. Units.....	37

1 BACNET ETHERNET DESCRIPTION

The BACnet Ethernet driver allows the FieldServer to transfer data to and from devices over Ethernet using BACnet Ethernet 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.

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

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

1.1 BTL Mark – BACnet Testing Laboratory



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

2 DRIVER SCOPE OF SUPPLY

2.1 Supplied by 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

Description
Ethernet 10/100 BaseT switch ²

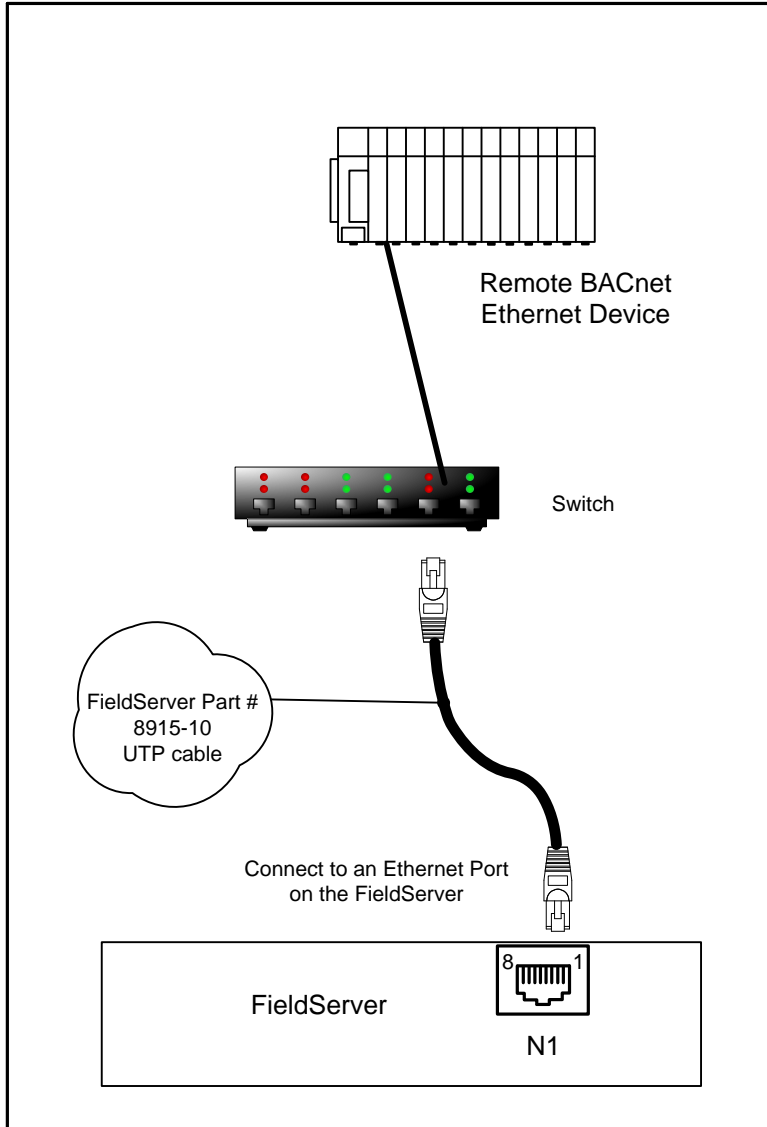
¹ This cable is necessary for connection to the driver. It is shipped with the FieldServer and not separately with the driver.

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

Example

```

// Data Arrays

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

5 CONFIGURING THE FIELDSEVER AS A BACNET ETHERNET CLIENT

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

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

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for BACnet Ethernet 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 must be mapped in the “Client Side Map Descriptors” section. Details on how to do this can be found below.

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

5.1 Client Side Connection Parameters

Section Title		
Connection		
Column Title	Function	Legal Values
Adapter	Adapter Name	N1, N2 ⁴
Protocol	Specify protocol used	BACnet_Eth
Poll_Delay*	Time between internal polls	0-32000s; 0.25s

Example

```

// Client Side Connections
Connections
Adapter      , Protocol      , Poll_Delay
N1           , Bacnet_Eth   , 0500s
    
```

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	1 - 16777215
Protocol	Specify protocol used	BACnet_Eth
Adapter	Specify port Adapter used	N1, N24

Example

```

// Client Side Nodes
Nodes
Node_Name   , Node_ID   , Protocol   , Adapter
DEV_01     , 1         , Bacnet_Eth , N1
    
```

⁴ Consult the appropriate Instruction manual for details of the ports available on specific hardware.

5.3 Client Side Map Descriptor Parameters

5.3.1 FieldServer Related 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 .	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 Related 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 5.2 .
Object_Type (Alias = Data_Type)	Type of object. Refer to Appendix D.1 for more information.	AI, AO, AV, BI, BO, BV, MI, MO, MV, NC
Object_Instance (Alias = Address)	Instance of the object on the device.	0, 1, 2, 3, ...4194303
Property	The BACnet property to be read.	Refer to 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 Appendix A.3 .	0 to max number of array items in the BACnet Property, -
Write_Priority*	Allows the driver to specify the write priority used to write an output.	1..16, 16
Linked_Map_Descriptors*	This parameter allows the linking of Map Descriptors to an active read or write Map Descriptor in order to construct a ReadPropertyMultiple or WritePropertyMultiple transaction. Refer to Appendix A.7.6 .	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.7.6

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 Examples

Client Side Map Descriptors

Map_Descriptors					
Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_Name	Object_Type
CMD_AI_01	, DA_AI_01	, 0	, Rdbc	, DEV_01	, AI
CMD_AI_02	, DA_AI_01	, 1	, Rdbc	, DEV_01	, AI
CMD_AI_03	, DA_AI_01	, 2	, Rdbc	, DEV_01	, AI
CMD_AO_01	, DA_AO_01	, 0	, Rdbc	, DEV_01	, AO
CMD_AO_02	, DA_AO_01	, 1	, Rdbc	, DEV_01	, AO
CMD_AO_03	, DA_AO_01	, 2	, Rdbc	, DEV_01	, AO

, Object_ID	, Property	, Scan_Interval
, 1	, Present_Value	, 20.000s
, 2	, Present_Value	, 20.000s
, 3	, Present_Value	, 20.000s
, 1	, Present_Value	, 30.000s
, 2	, Present_Value	, 30.000s
, 3	, Present_Value	, 30.000s

Map_Descriptors					
Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_Name	Object_Type
CMD_DI_01	, DA_DI_01	, 0	, Rdbc	, DEV_01	, BI
CMD_DI_02	, DA_DI_01	, 1	, Rdbc	, DEV_01	, BI
CMD_DI_03	, DA_DI_01	, 2	, Rdbc	, DEV_01	, BI
CMD_DO_01	, DA_DO_01	, 0	, Rdbc	, DEV_01	, BO
CMD_DO_02	, DA_DO_01	, 1	, Rdbc	, DEV_01	, BO
CMD_DO_03	, DA_DO_01	, 2	, Rdbc	, DEV_01	, BO

, Object_ID	, Property	, Scan_Interval
, 1	, Present_Value	, 15.000s
, 2	, Present_Value	, 15.000s
, 3	, Present_Value	, 15.000s
, 1	, Present_Value	, 30.000s
, 2	, Present_Value	, 30.000s
, 3	, Present_Value	, 30.000s

6 CONFIGURING THE FIELDSEVER AS A BACNET ETHERNET SERVER

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

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

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

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

6.1 Server Side Connection Parameters

Section Title		
Connections		
Column Title	Function	Legal Values
Adapter	Adapter Name	N1, N2 ⁵
Protocol	Specify protocol used	BACnet_Eth
Virtual_Network_Number*	Specify a unique network number if there are multiple virtual Server Nodes. Refer to Appendix A.2 .	1 – 65535, 5

Example

```
// Server Side Connections
Connections
Adapter      , Protocol6    , Virtual_Network_Number
N1           , Bacnet_Eth  , 5
```

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.

⁶ No connection information is necessary on Server side.

6.2 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.	1 - 16777215
Protocol	Specify protocol used.	BACnet_Eth
Node_Option*	Enable or disable COV for this Node. Refer to Appendix A.7 for further information.	COV_Enable, COV_Disable, -
Node_Type*	Specific to Trane applications. The Node type can be defined. Refer to Appendix C.1 .	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.11 for more information.	-, WPM, COV
Model_Name*	Sets the Model Name Property of the Device Object.	Up to 32 alphanumeric characters

Example

```
// Server Side Nodes

Nodes
Node_Name      , Node_ID  , Protocol    , Node_Option  , Model_Name
Virtual_DEV_11 , 1          , Bacnet_Eth , COV_Enable   , BACnet Ethernet
```

6.3 Server Side Map Descriptor Parameters

6.3.1 FieldServer Related 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 .	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

6.3.2 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 6.2 .
Object_Type (Alias = Data_Type)	Data type in Controller.	AI, AO, AV, BI, BO, BV, MI, MO, MV, NC. Refer to Appendix D.1 .
Object_Instance (Alias = Address)	Instance of the Object on the Device.	Integers between 0 to 4194303
Units*	The object units.	Refer to Appendix D.4, m2
Data_Array_Low_Scale*	Scaling zero in Data Array.	-32767 to 32767, 0
Data_Array_High_Scale*	Scaling max in Data Array.	-32767 to 32767, 100
Node_Low_Scale*	Scaling zero in Connected Node.	-32767 to 32767, 0
Node_High_Scale*	Scaling max in Connected Node.	-32767 to 32767, 100
Active_Text 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. Refer to Appendix A.10.1 .	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.7 .	One of the configured Mapdescriptor_Names of type NC (Notification Class). Objects are not available to any NC if left out.
Ack_Required*	For a Notification_Class Object, specify whether EventNotifications require a user Acknowledgement. Refer to Appendix A.7 .	Yes, No
COV_Increment*	For a Server Map Descriptor of type AO or AI, initialize the COV_Increment property. Refer to Appendix A.7 .	Any Float value, 0
Input_alarm_State*	For binary points the user can specify which of the states (0 or 1) to regard as an alarm state.	0, 1
Description*	Specify the object's description property. E.g. Room Temp. Refer to Appendix A.7.3 .	Any text string of length up to 40 characters, If not configured, defaults to Object_Name.

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

Section Title		
Map Descriptors		
Column Title	Function	Legal Values
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 Appendix A.10 .	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.7.7 .	Map Descriptor defined per Appendix A.7 .

6.3.3 Map Descriptor Example

```
// Server Side Map Descriptors

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name
SMD_AI_01 , DA_AI_01 , 0 , Passive , Virtual_DEV_11
SMD_AI_02 , DA_AI_01 , 1 , Passive , Virtual_DEV_11
SMD_AI_03 , DA_AI_01 , 2 , Passive , Virtual_DEV_11
SMD_AO_01 , DA_AO_01 , 0 , Passive , Virtual_DEV_11
SMD_AO_02 , DA_AO_01 , 1 , Passive , Virtual_DEV_11
SMD_AO_03 , DA_AO_01 , 2 , Passive , Virtual_DEV_11
```

Object_Type	Object_Instance	Units
AI	1	Degrees-Fahrenheit
AI	2	Degrees-Fahrenheit
AI	3	Degrees-Fahrenheit
AO	1	percent-relative-humidity
AO	2	percent-relative-humidity
AO	3	percent-relative-humidity

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

Appendix A. Useful Features

Appendix A.1. BACnet Object Names

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

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

Appendix A.2. Virtual Network Number

Whenever there are multiple virtual nodes, the FieldServer interconnects these on a virtual internal network and acts as a virtual router between this and the external network. Each internal virtual network must be assigned a network number. The BACnet specification states that this network number must be unique across the entire BACnet internetwork. The following configuration error is generated when an attempt is made to reuse the same network number on the same device:

```
BACnet Router : Error: Network Number 5 has already been used. Configure a unique Virtual_Network_Number on each BACnet _MSTP Connection
```

NOTE: Failing to use unique network numbers can lead to routing problems on a BACnet network.

The Virtual_Network_Number parameter can be used to assign network numbers to the internal virtual networks associated with each BACnet connection. The defaults will also be unique for the FieldServer itself, but it is up to the user to ensure that the network number is unique across the entire network. The error message shown above only arises when the same virtual network number is specified for more than one connection.

Section Title		
Connections		
Column Title	Function	Legal Values
Virtual_Network_Number*	Specify a unique network number if there are multiple virtual Server nodes.	1-65534, 5

Example

```
// Server Side Connections
Connections
Adapter      , Protocol      , Virtual_Network_Number
N1           , Bacnet_Eth   , 5
```

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

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

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

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

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

Map_Descriptors					
Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_Name	Object_Type
CMD_AI_01	DA_AI_01	0	Rdbc	DEV_01	AO

Object_ID	Property	Array_Index
1	Priority_Array	7

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

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

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

- When an Array_Index is specified, the ReadPropertyMultiple or WritePropertyMultiple request will iterate through N successive Array_Index values for the given Object and Property (e.g. Priority Array Index 1 to 16 of Analog Output 1)
- When no Array_Index is specified, the ReadPropertyMultiple or WritePropertyMultiple request will iterate through N successive Object Instances, reading the specified Property from each successive object (e.g. Present Value of Analog Output 1 to Analog Output 16)

Appendix A.4. FieldServer Implementation of BACnet Priority Arrays

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

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

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

Appendix A.4.1. Accessing Priority Array information

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

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

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

, Length , Relinquish_default , DA_Pri_Array , DA_Pri_Array_Offset , DA_Pri_In_Use , DA_Pri_In_Use_Offset
, 1 , 40.56 , DA_Pri_Array_1 , 0 , DA_Pri_in_use_1 , 0
```

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

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

Map_Descriptors						
Map_Descriptor_Name	Data_Type	Function	Scan_Interval	Data_Array_Name	Data_Array_Index	Node_Name
CMD AO	AO	Rdbc	1.0s	DA AO	2	N1 1
CMD AO Rel	AO	Wrbx	1.0s	DA AO	3	N1 1

Address	Length	Write_Priority	Service
1	1	7	-
1	1	7	Relinquish

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

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: The length of the Address must be 1.

Appendix A.6. BACnet State Text Preload

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

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

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

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

```
//set up a look up table
```

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

NOTE: Valid entries for Table_Index_Value are integers of 1 to 16.

Data_Arrays		
Data_Array_Name	Data_Format	Data_Array_Length
DA_MI_01	UINT16	100

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

Object_Type	Object_ID	Length	State_Text_Array
MI	1	1	Fire_Alm_Text

Appendix A.6.2. Method 2 – Using a Single Data Array

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

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

```
Map_Descriptors
Map_Descriptor_Name , Object_Type , Object_Instance , Function , Data_Array_Name
CMD_MI_01           , MI           , 1               , Passive , DA_MI_01
```

```
 , Node_Name , Length , State_Text_Array
 , N1 11     , 1      , Da_State_Txt
```

Appendix A.6.3. Using Intrinsic Reporting for a Multistate Value

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

- 0 = normal
- 1 = alarm
- 2 = fault

```
//set up a look up table

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

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

Appendix A.7. COV and Intrinsic Reporting

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

The services are suited to different purposes:

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

Appendix A.7.1. COV Configuration Tips

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

Appendix A.7.2. Intrinsic Reporting Configuration Tips

- 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 not stored permanently but are kept in volatile memory, and subscriptions must be renewed on system restart.
- A permanent static RecipientList may be configured directly on the FieldServer, eliminating the need for a remote workstation to modify the RecipientList in order to receive Event Notifications. Using this option makes the RecipientList read-only, i.e. it can then only be modified via the FieldServer config.csv file. Refer to [Appendix A.7.7](#).
- 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 such as if it is set up with a reference to a Notification Class map descriptor.

Appendix A.7.3. Map Descriptor Example - Enable or Disable COV

Nodes			
Node_Name	Node_ID	Protocol	Node_Option
Virtual_Dev_11	11	Bacnet_Eth	COV_Disable

Appendix A.7.4. Map Descriptor Example - COV

```
// Notification Class Object
```

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

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

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

Appendix A.7.5. Map Descriptor Example – Intrinsic Reporting

```
// Notification class Objects
```

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

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

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

Appendix A.7.6. Setting the Priority of Intrinsic Alarming

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

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

Priority Values should be chosen as follows:

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

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

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

This is done by adding an Event_Receiver section to the config.csv. This section must appear after the definition and all other references to the relevant Notification Class object. The example below is the Notification Class map descriptor governing the sending of Event Notifications (Intrinsic Alarming).

```
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
```

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. The example below is a BACnet Object that will report its alarms via the Notification Class map descriptor BAC_NC_P1 (example above).

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

The example below sets up the 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.

```
Event_Receiver
Node_Name , Node_ID , Protocol , Process_ID , Notification_Class_MD , Adapter
OWS_01 , 10 , BACnet_Eth , 12345 , BAC_NC_P1 , N1
OWS_02 , 12 , BACnet_Eth , 56789 , BAC_NC_P1 , N1
```

Appendix A.8. Factors Determining the Reliability Property

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

- For Multistate Inputs, Outputs and Values, the Reliability property is set to MULTI_STATE_FAULT (9) when the Present_Value of the object corresponds to an entry in the Fault_Values property of the object, as configured by the State_Text table in the configuration file.
- For all object types, the Reliability property is set to COMMUNICATIONS_FAILURE (12) when the responsible Server Node is offline.
- When neither Condition 1 nor Condition 2 is met, the Reliability property is set to NO_FAULT_DETECTED (0).

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

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

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

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

For example:

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


Config File Examples:

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

Map_Descriptor_Name – 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.

Map_Descriptor_Name	Object_Type	Property	Function	Data_Array_Name	Data_Array_Offset
CMD WPM 1	, AO	, -	, Passive_Client	, DA WPM	, 3
CMD WPM 2	, AO	, -	, Passive_Client	, DA WPM	, 7
CMD WPM 3	, Device	, Max_Master	, Passive_Client	, DA WPM	, 12
CMD WPM 4	, Device	, Max_Info_Frames	, Passive_Client	, DA WPM	, 13
CMD WPM 5	, BO	, -	, Passive_Client	, DA BOP	, 0

Node_Name	Address	Length	Write_Priority	Linked_Map_Descriptor
, N1 1	, 4	, 3	, 8	, CMD WPM
, N1 1	, 8	, 4	, 10	, CMD WPM
, N1 1	, 1	, 1	, 10	, CMD WPM
, N1 1	, 1	, 1	, 10	, CMD WPM
, N1 1	, 0	, 10	, 11	, CMD WPM

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

Function – Set the function to Passive_Client.

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

Appendix A.10. Update Property Function

The Update_Property parameter may be configured on a BACnet Server Map Descriptor in order to allow a property other than the Present_Value to be updated in addition to the Present_Value when the Data Array value changes.

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

Appendix A.10.1. Supported Values

- Present_Value (default operation)
- Relinquish_Default

Appendix A.10.2. Mode of Operation when Used for the Relinquish_Default Property

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

Instead, it is possible to use the Update_Property setting to initialize the Relinquish_Default property using the actual field Present_Value read from the Client Side device, so that a restart achieves two important outcomes:

- The Client side outputs are not modified.
- The current state of the Client Side output can be read via the Relinquish_Default property. This is especially useful since the BACnet specification makes no provision for Feedback_Values on analog objects.

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

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

Restrictions:

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

Appendix A.10.3. Update Property Configuration Example

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

Appendix A.11. Disabling Selected BACnet Services

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

- COV - certain BACnet MS/TP routers struggle to keep up with the bidirectional traffic that can result from many COVSubscribe requests in quick succession. In this case, better results might be obtained by turning COV off.
- WritePropertyMultiple (WPM) - when the user requires writes to the BACnet Server to be acknowledged only once Client Side writes have succeeded, WPM is turned off.

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

The following values may be used under Disabled_Services:

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

Examples:

Disable WPM only

```
Nodes
Node_Name , Node_ID , Protocol , Disabled_Services
NN 01 , 1 , BACnet_Eth , WPM
```

Disable COV only

```
Nodes
Node_Name , Node_ID , Protocol , Disabled_Services
NN 02 , 2 , BACnet_Eth , COV
```

Disable WPM and COV

```
Nodes
Node_Name , Node_ID , Protocol , Disabled_Services
NN 03 , 3 , BACnet_Eth , WPM COV
```

Appendix A.12. Using a .ini File to Set the Vendor_ID and Vendor_Name of the Device Object

The vendor.ini file can be used to change the BACnet Vendor Name and Vendor ID if desired. A file with the following format must be created and downloaded to the FieldServer using the FS-GUI Update Files function by clicking Setup, then File Transfer in the Navigation tree. Then go to the General tab and choose the vendor.ini file.

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

NOTE: Changing the Vendor_ID and Vendor_Name on FieldServer BTL approved products may disqualify them from the BTL approval.

Appendix A.13. Writing to Custom Properties on Remote BACnet Devices

The Custom_Property and Tag_Type parameters allow the FieldServer to write to devices that implemented custom properties that are not part of the BACnet protocol specification.

The following example writes to custom property 650 with Tag_Type 1, 9 and 4 (which are examples vendor specific properties):

Map_Descriptors					
Map_Descriptor_Name	Node_Name	Function	Data_Array_Name	Data_Array_Index	Data_Type
MultistateWR	, BCU_12	, WRBX	, DA_MV	, 1	, MV
BinaryWR	, BCU_12	, WRBX	, DA_BV	, 1	, BV
FloatWR	, BCU_12	, WRBX	, DA_AV	, 1	, AV

, Object_ID	, Custom_Property	, Tag_Type	, Length
, 1	, 650	, 1	, 1
, 1	, 650	, 9	, 1
, 1	, 650	, 4	, 1

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 slave, ensure the FieldServer's Subnet Mask matches the Subnet Mask of the BACnet Master. Otherwise, communications are very slow and eventually stop altogether.

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.	This message should not appear. 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 Eth DLL Temporarily out of receive buffers	The FieldServer was flooded with more BACnet Eth 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:

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

BACnet Error CODES for Error Class OBJECT	
Other	0
Authentication failed	1
Configuration in progress	2
Device busy	3
Dynamic creation not supported	4
File access denied	5
Incompatible security levels	6
Inconsistent parameters	7
Inconsistent selection criterion	8
Invalid data type	9
Invalid file access method	10
Invalid file start position	11
Invalid operator name	12
Invalid parameter data type	13
Invalid time stamp	14
Key generation error	15
Missing required parameter	16
No objects of specified type	17
No space for object	18
No space to add list element	19
No space to write property	20
No vt sessions available	21

BACnet Error CODES for Error Class OBJECT	
Property is not a list	22
Object deletion not permitted	23
Object identifier already exists	24
Operational problem	25
Password failure	26
Read access denied	27
Security not supported	28
Service request denied	29
Timeout	30
Unknown object	31
Unknown property	32
Unknown vt class	34
Unknown vt session	35
Unsupported object type	36
Value out of range	37
Vt session already closed	38
Vt session termination failure	39
Write access denied	40
Character set not supported	41
Invalid array index	42
Invalid index	42

Appendix C. Vendor Information

Appendix C.1. McQuay

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

Appendix C.2. Trane

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

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

Appendix C.2.1. Enabling Trane Alarming

The Driver will automatically support Trane Alarming if the Node_Type parameter as defined in **Section 6.2** 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 6.2 .
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	
ABB_vfd1_Output Frequency	ai1vfd1	, 1	, Passive	, Bac11	, AI	
ABB_vfd1_Current	ai1vfd1	, 2	, Passive	, Bac11	, AI	

, Object_ID	, Length	, Event_Receiver_Name	, Low_Alarm	, High_Alarm	, Event_Class	, Ack_Required	, Alarm_Deadband
, 1	, 1	, ADD_WS	, 20	, 30	, 5	, Yes	, 2
, 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	
ABB_VFd1_RO_1 Status	bi1vfd1	, 0	, Passive	, Bac11	, BI	, 1	
ABB_VFd1_Run Status	bi1vfd1	, 1	, Passive	, Bac11	, BI	, 2	

, Length	, Event_Receiver_Name	, Input_Alarm_State	, Event_Class	, Ack_Required
, 1	, ADD_WS	, 1	, 5	, Yes
, 1	, ADD_WS	, 1	, 5	, Yes

Appendix C.3. Liebert

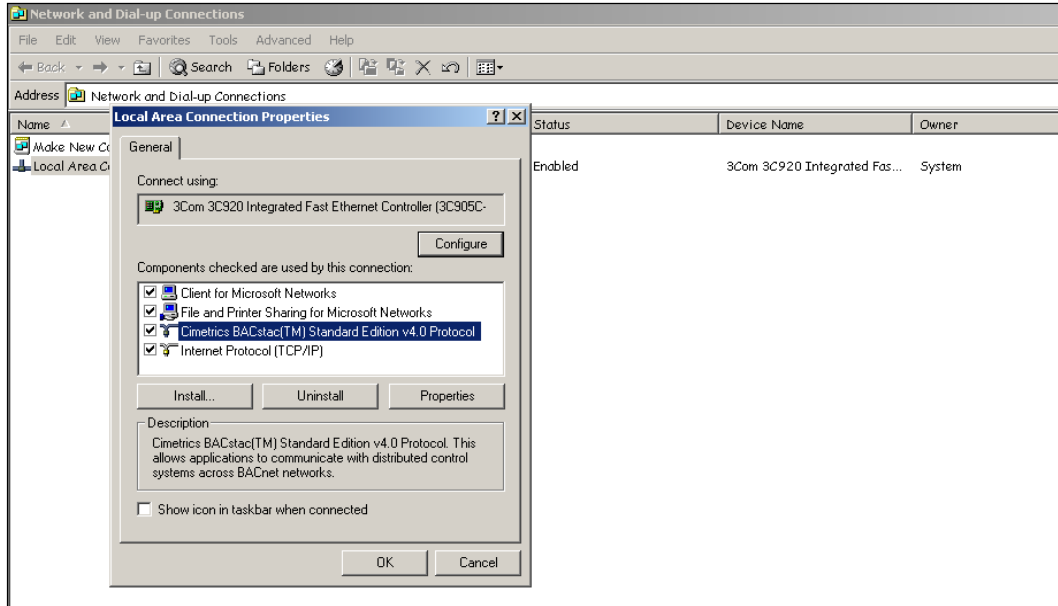
Polling BACnet addresses that are not configured for Liebert systems may cause the connection to fail in older versions of Liebert. Contact your Liebert supplier for more information.

Appendix C.4. Honeywell EBI

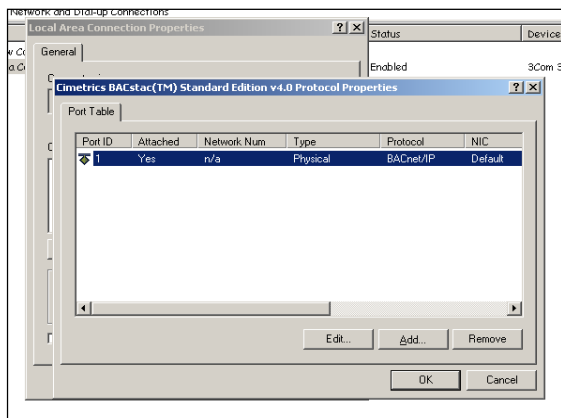
Honeywell EBI cannot process EventNotifications with ACK_Required set to 1. The ACK_Required property of Notification Class Map Descriptors configured for use with EBI must therefore be set to 0.

Appendix C.5. Using Cimetrics Explorer

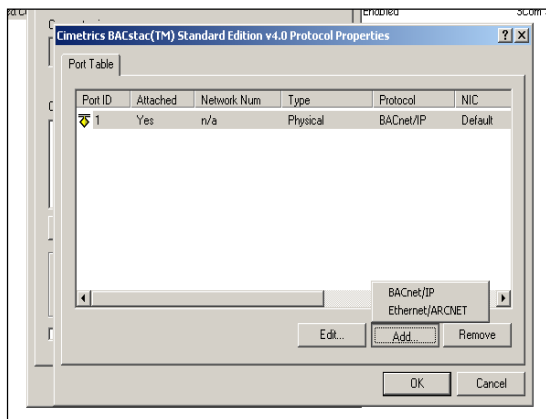
Cimetrics Explorer needs to be configured to use either BACnet Ethernet or BACnet/IP. This is not a setting in the actual Cimetrics BACnet Explorer, but is done in the Cimetrics BACstac Protocol that can be found in the Local Area Connection Properties as shown below:



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 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

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

Appendix D.3. 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.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.

Appendix D.4. Units

Unit	Variation 1	Variation 2	Variation 3
Area			
square-centimeters			
square-feet			
square-inches			
square-meters			
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			
farads			
henrys			
ohms			
ohm-meters			
kilohms			
megohms			
micro_siemens			
siemens			
siemens-per-meter			
TESLAS			
Volts	Voltage		
Voltage			
Millivolts			
kilovolts			
megavolts			
Volt-Amperes	Volt-Amps	VA	
kilovolt-amperes	kilovolt-amps	KVA	
megavolt-amperes	megavolt-amps		
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			

Unit	Variation 1	Variation 2	Variation 3
Energy			
Joules			
kilojoules			
kilojoules-per-kilogram			
megajoules			
watt-hours	Wh		
kilowatt-hours	KWh		
Megawatt-hours	MWh		
BTUs			
Kilo-BTUs			
Mega-BTUs			
Therms			
ton-hours			
kilovolt-amperes--hours			
kilovolt-amperes-reactive-hours			
megavolt-amperes--hours			
megavolt-amperes-reactive-hours			
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			
millimeters			
centimeters			
meters			
inches			
feet			
Light			
candelas			
candelas-per-square-meter			
watts-per-square-foot			
watts-per-square-meter			
lumens			
luxes			
foot-candles			
Mass			
Kilograms	Kg		
Pounds-mass			
Tons			

Unit	Variation 1	Variation 2	Variation 3
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			
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	
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			

Unit	Variation 1	Variation 2	Variation 3
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			
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-min			
liters-per-second			
liters-per-minute			
liters-per-hour			
US-Gallons-per-hour			
US-Gallons-per-minute	GPM		
Others			
degrees-angular			
degrees-celsius-per-hour			
degrees-celsius-per-minute			
degrees-fahrenheit-per-hour			
degrees-fahrenheit-per-minute			
joule-seconds			
kilogram-per-cubic-meter			
kilowatt-hour-per-square-meter			
kilowatt-hour-per-square-foot			
megajoules-per-square-meter			
megajoules-per-square-foot			
No-Units	No Units	No_Units	None
newton-seconds			
newtons-per-meter			
parts-per-million			
parts-per-billion			
Percent			
percent-obscuration-per-foot			
percent-obscuration-per-meter			
percent-per-second			
per-minute			
per-second			
psi-per-degrees-fahrenheit			
radians			
revolutions-per-minute			
square-meters-per-newton			
watts-per-meter-per-degree-kelvin			
watts-per-square-meter-degrees-kelvin			