BACNE Communication Driver

Driver for ethernet communication with devices using **B**uilding **A**utomation and **C**ontrol **Net**work (BACnet)/IP protocol

Contents

INTRODUCTION **3**

GENERAL INFORMATION **4**

- Device Specifications 4
- Network Specifications 4
- Driver Characteristics 4
- Analog Input Object (AI) 5
- Analog Output Object (AO) 6
- Analog Value Object (AV) 8
- Binary Input Object (BI) 9
- Binary Output Object (BO) 11
- Binary Value Object (BV) 13
- Multi-State Input Object (MSI) 14
- Multi-State Output Object (MSO) 16
- Multi-State Value Object (MSV) 17
- Schedule Object (SCH) 18
- Device Object (DEV) 19
- Life Safety Point Object (LSP) 20
- Life Safety Zone Object (LSZ) 22
- Accumulator Object (ACC) 24
- Averaging Object (AVR) 26
- Calendar Object (CAL) 27
- Command Object (CMD) 28
- Event Enrollment Object (EVT) 29
- Loop Object Type (LOP) 31
- Notification Class Object Type (NOT) 35
- Program Object Type (PRG) 36
- Pulse Converter Object Type (PC) 38
- Group Object Type (GRP) 42
- File Object Type (FO) 42
- Trend Log Object Type (TL) 43
- Conformance Testing 46

CONFIGURING THE DEVICE **48**

SELECTING THE DRIVER **49**

CONFIGURING THE DRIVER **50**

- Configuring the Communication Settings 50
- Configuring the Driver Station 52
- Configuring the Driver Worksheets 54

EXECUTING THE DRIVER **71**

TROUBLESHOOTING **72**

SAMPLE APPLICATION 74

REVISION HISTORY........ 75

Introduction

The BACNE driver enables communication between the Studio system and devices using the BACnet/IP protocol over UDP/IP, according to the specifications discussed in this document.

This document will help you to select, configure and execute the BACNE driver, and it is organized as follows:

- **Introduction**: This section, which provides an overview of the document.
- **General Information**: Identifies all of the hardware and software components required to implement communication between the Studio system and the target device.
- **Selecting the Driver**: Explains how to select the BACNE driver in the Studio system.
- **Configuring the Device**: Describes how the target device must be configured to receive communication from the BACNE driver.
- **Configuring the Driver**: Explains how to configure the BACNE driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver**: Explains how to execute the BACNE driver during application runtime.
- **Troubleshooting**: Lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.
- **Sample Application**: Explains how to use a sample application to test the BACNE driver configuration.
- **Revision History**: Provides a log of all changes made to the driver and this documentation.

Notes:

- This document assumes that you have read the “Development Environment” chapter in Studio’s *Technical Reference Manual*.
- This document also assumes that you are familiar with the Microsoft Windows environment. If you are not familiar with Windows, then we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.

General Information

This chapter identifies all of the hardware and software components required to implement Ethernet communication between the BACNE driver in Studio and a device using BACnet/IP protocol over UDP/IP.

The information is organized into the following sections:

- Device Specifications
- Network Specifications
- Driver Characteristics

Device Specifications

To establish communication, your target device must meet the following specifications:

- **Compatible Equipment**: All devices that are fully compatible with BACnet/IP protocol over UDP/IP

Network Specifications

To establish communication, your device network must meet the following specifications:

- **Physical Protocol**: Ethernet
- **Logic Protocol**: BACnet/IP
- **Device Runtime Software**: None

Driver Characteristics

The BACNE driver package consists of the following files, which are automatically installed in the **`\DRV`** subdirectory of Studio:

- **`BACNE.INI:`** Internal driver file. *You must not modify this file*.
- **`BACNE.MSG:`** Internal driver file containing error messages for each error code. *You must not modify this file*.
- **`BACNE.PDF:`** This document, which provides detailed information about the BACNE driver.
- **`BACNE.DLL:`** Compiled driver.

You can use the BACNE driver on the following operating systems:

- Windows 7, 8
- Windows CE, Embedded

The BACNE driver supports the following objects:

Analog Input Object (AI)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0
PRESENT-VALUE	float	123.2
DESCRIPTION	string	This sensor monitors the core activity.
DEVICE-TYPE	integer	5
STATUS-FLAGS	string 4 Boolean values encoded as string (1 = true and 0 = false) {IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}	1010
EVENT-STATE	integer	0
RELIABILITY	integer	0
OUT-OF-SERVICE	integer	0
UPDATE-INTERVAL	integer	0
UNITS	integer	0
MIN-PRES-VALUE	integer	0
MAX-PRES-VALUE	integer	0
RESOLUTION	float	0.1
COV-INCREMENT	integer	0
TIME-DELAY	integer	0
NOTIFICATION-CLASS	integer	0
HIGH-LIMIT	integer	0

LOW-LIMIT	integer	0
DEADBAND	integer	0
LIMIT-ENABLE	string 2 Boolean values (1 = true and 0 = false) {lowLimitEnable, highLimitEnable}	01
EVENT-ENABLE	string 3 Boolean values (1 = true and 0 = false) {to-offnormal, to-fault, to-normal}	101
ACKED-TRANSITIONS	string 3 Boolean values (1 = true and 0 = false) {TO-OFFNORMAL, TO-FAULT, TO-NORMAL}	101
NOTIFY-TYPE	integer	1
EVENT-TIME-STAMPS	string Sequence of time stamps.	2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0
PROFILE-NAME	string	123-AI

Analog Output Object (AO)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0
PRESENT-VALUE	float	123.2
DESCRIPTION	string	This sensor monitors the core activity.
DEVICE-TYPE	Integer	5
STATUS-FLAGS	string 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	1010
EVENT-STATE	integer	0

RELIABILITY	integer	0
OUT-OF-SERVICE	integer	0
UNITS	integer	0
MIN-PRES-VALUE	integer	0
MAX-PRES-VALUE	integer	0
RESOLUTION	real	0.1
PRIORITY-ARRAY	string Sequence of **16** values separated by "|". The type of these values can be one of the following (**Null**, **real**, **binary**, **integer**) For more information look at ***BACnetPriorityArray*** type on the data types section.	NULL|NULL|NULL|NULL|NULL|NULL|12|NULL| NULL|NULL|NULL|NULL|NULL|NULL|NULL|25.000000
RELINQUISH-DEFAULT	real	12.3
COV-INCREMENT	integer	0
TIME-DELAY	integer	0
NOTIFICATION-CLASS	integer	0
HIGH-LIMIT	integer	0
LOW-LIMIT	integer	0
DEADBAND	integer	0
LIMIT-ENABLE	string 2 Boolean values (1 = true and 0 = false) **{lowLimitEnable, highLimitEnable}**	01
EVENT-ENABLE	string 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	101
ACKED-TRANSITIONS	string 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	101
NOTIFY-TYPE	integer	1
EVENT-TIME-STAMPS	string Sequence of time stamps.	2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0
PROFILE-NAME	string	Test

Analog Value Object (AV)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0
PRESENT-VALUE	float	123.2
DESCRIPTION	string	This sensor monitors the core activity.
STATUS-FLAGS	string 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	1010
EVENT-STATE	integer	0
RELIABILITY	integer	0
OUT-OF-SERVICE	integer	0
UNITS	integer	0
PRIORITY-ARRAY	string Sequence of **16** values separated by "|". The type of these values can be one of the following (**Null**, **real**, **binary**, **integer**) For more information look at ***BACnetPriorityArray*** type on the data type section.	NULL|NULL|NULL|NULL|NULL|NULL|12|NULL| NULL|NULL|NULL|NULL|NULL|NULL|NULL|25.000000
RELINQUISH-DEFAULT	real	12.3
COV-INCREMENT	integer	0
TIME-DELAY	integer	0
NOTIFICATION-CLASS	integer	0
HIGH-LIMIT	integer	0
LOW-LIMIT	integer	0
DEADBAND	integer	0
LIMIT-ENABLE	string	01

	2 Boolean values (1 = true and 0 = false) **{lowLimitEnable, highLimitEnable}**	
EVENT-ENABLE	string 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	101
ACKED-TRANSITIONS	string 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	101
NOTIFY-TYPE	integer	1
EVENT-TIME-STAMPS	string Sequence of time stamps.	2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0
PROFILE-NAME	string	Test

Binary Input Object (BI)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0
PRESENT-VALUE	integer	123
DESCRIPTION	string	This sensor monitors the core activity.
DEVICE-TYPE	integer	5
STATUS-FLAGS	string 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	1010
EVENT-STATE	integer	0
RELIABILITY	integer	0

OUT-OF-SERVICE	integer	0
POLARITY	integer **{normal = 0, reverse = 1}**	1
INACTIVE-TEXT	string	test
ACTIVE-TEXT	string	Test
CHANGE-OF-STATE-TIME	string Date and Time separated by a space character For more information look at ***BACnetDateTime*** type on the data type section.	23-8-2011 9:42:44.64
CHANGE-OF-STATE-COUNT	unsigned integer	33
TIME-OF-STATE-COUNT-RESET	string Date and Time separated by a space character For more information look at ***BACnetDateTime*** type on the data type section.	23-8-2011 9:42:44.64
ELAPSED-ACTIVE-TIME	string Date and Time separated by a space character For more information look at ***BACnetDateTime*** type on the data type section.	23-8-2011 9:42:44.64
TIME-OF-ACTIVE-TIME-RESET	string Date and Time separated by a space character For more information look at ***BACnetDateTime*** type on the data type section.	23-8-2011 9:42:44.64
TIME-DELAY	unsigned integer	10
NOTIFICATION-CLASS	integer	0
ALARM-VALUE	integer	1
EVENT-ENABLE	string 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	101
ACKED-TRANSITIONS	string 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	101
NOTIFY-TYPE	integer	1

EVENT-TIME-STAMPS	string Sequence of time stamps.	2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0
PROFILE-NAME	string	Test

Binary Output Object (BO)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0
PRESENT-VALUE	integer	123
DESCRIPTION	string	This sensor monitors the core activity.
DEVICE-TYPE	integer	5
STATUS-FLAGS	string 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	1010
EVENT-STATE	integer	0
RELIABILITY	integer	0
OUT-OF-SERVICE	integer	0
POLARITY	integer **{normal = 0, reverse = 1}**	1
INACTIVE-TEXT	string	test
ACTIVE-TEXT	string	Test
CHANGE-OF-STATE-TIME	string Date and Time separated by a space character For more information look at ***BACnetDateTime*** type on the data type section.	23-8-2011 9:42:44.64
CHANGE-OF-STATE-COUNT	unsigned integer	33

TIME-OF-STATE-COUNT-RESET	string Date and Time separated by a space character For more information look at ***BACnetDateTime*** type on the data type section.	23-8-2011 9:42:44.64
ELAPSED-ACTIVE-TIME	unsigned integer	12
TIME-OF-ACTIVE-TIME-RESET	string Date and Time separated by a space character For more information look at ***BACnetDateTime*** type on the data type section.	23-8-2011 9:42:44.64
MINIMUN-OFF-TIME	integer	0
MINIMUN-ON-TIME	integer	0
PRIORITY-ARRAY	string Sequence of **16** values separated by "|". The type of these values can be one of the following (**Null**, **real**, **binary**, **integer**) For more information look at ***BACnetPriorityArray*** type on the data type section.	NULL|NULL|NULL|NULL|NULL|NULL|12|NULL| NULL|NULL|NULL|NULL|NULL|NULL|NULL|25.000000
RELINQUISH-DEFAULT	real	12.3
TIME-DELAY	integer	0
NOTIFICATION-CLASS	integer	0
FEEDBACK-VALUE	integer **{inactive (0), active (1)}**	1
EVENT-ENABLE	string 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	101
ACKED-TRANSITIONS	string 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	101
NOTIFY-TYPE	integer	1
EVENT-TIME-STAMPS	string Sequence of time stamps.	2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0
PROFILE-NAME	string	Test

Binary Value Object (BV)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0
PRESENT-VALUE	integer	123
DESCRIPTION	string	This sensor monitors the core activity.
STATUS-FLAGS	string 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	1010
EVENT-STATE	integer	0
RELIABILITY	integer	0
OUT-OF-SERVICE	integer	0
INACTIVE-TEXT	string	test
ACTIVE-TEXT	string	Test
CHANGE-OF-STATE-TIME	string Date and Time separated by a space character For more information look at ***BACnetDateTime*** type on the data type section.	23-8-2011 9:42:44.64
CHANGE-OF-STATE-COUNT	unsigned integer	33
TIME-OF-STATE-COUNT-RESET	string Date and Time separated by a space character For more information look at ***BACnetDateTime*** type on the data type section.	23-8-2011 9:42:44.64
ELAPSED-ACTIVE-TIME	unsigned integer	12
TIME-OF-ACTIVE-TIME-RESET	string Date and Time separated by a space character	23-8-2011 9:42:44.64

	For more information look at ***BACnetDateTime*** type on the data type section.	
MINIMUN-OFF-TIME	integer	0
MINIMUN-ON-TIME	integer	0
PRIORITY-ARRAY	string Sequence of **16** values separated by "|". The type of these values can be one of the following (**Null**, **real**, **binary**, **integer**) For more information look at ***BACnetPriorityArray*** type on the data type section.	NULL|NULL|NULL|NULL|NULL|NULL|12|NULL| NULL|NULL|NULL|NULL|NULL|NULL|NULL|25.000000
RELINQUISH-DEFAULT	real	12.3
TIME-DELAY	integer	0
NOTIFICATION-CLASS	integer	0
ALARM-VALUE	integer	1
EVENT-ENABLE	string 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	101
ACKED-TRANSITIONS	string 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	101
NOTIFY-TYPE	integer	1
EVENT-TIME-STAMPS	string Sequence of time stamps.	2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0
PROFILE-NAME	string	Test

Multi-State Input Object (MSI)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0

PRESENT-VALUE	float	123.2
DESCRIPTION	string	This sensor monitors the core activity.
DEVICE-TYPE	integer	5
STATUS-FLAGS	string 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	1010
EVENT-STATE	integer	0
RELIABILITY	integer	0
OUT-OF-SERVICE	boolean	0
NUMBER-OF-STATES	unsigned integer	2
STATE-TEXT	string	test
TIME-DELAY	integer	0
NOTIFICATION-CLASS	integer	0
ALARM-VALUES	string **List of** unsigned integer.	1|1|2
FAULT-VALUES	string **List of** unsigned integer.	1|3
EVENT-ENABLE	string 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	101
ACKED-TRANSITIONS	string 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	101
NOTIFY-TYPE	integer	1
EVENT-TIME-STAMPS	string Sequence of time stamps.	2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0
PROFILE-NAME	string	Test

Multi-State Output Object (MSO)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0
PRESENT-VALUE	float	123.2
DESCRIPTION	string	This sensor monitors the core activity.
DEVICE-TYPE	integer	5
STATUS-FLAGS	string 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	1010
EVENT-STATE	integer	0
RELIABILITY	integer	0
OUT-OF-SERVICE	boolean	0
NUMBER-OF-STATES	unsigned integer	2
STATE-TEXT	string	test
PRIORITY-ARRAY	string Sequence of **16** values separated by "|". The type of these values can be one of the following (**Null**, **real**, **binary**, **integer**) For more information look at ***BACnetPriorityArray*** type on the data types section.	NULL|NULL|NULL|NULL|NULL|NULL|12|NULL| NULL|NULL|NULL|NULL|NULL|NULL|NULL|25.000000
RELINQUISH-DEFAULT	real	12.3
TIME-DELAY	integer	0
NOTIFICATION-CLASS	integer	0
FEEDBACK-VALUE	integer **{inactive (0), active (1)}**	1
EVENT-ENABLE	string 3 Boolean values	101

	(1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	
ACKED-TRANSITIONS	string 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	101
NOTIFY-TYPE	integer	1
EVENT-TIME-STAMPS	string Sequence of time stamps.	2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0
PROFILE-NAME	string	Test

Multi-State Value Object (MSV)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0
PRESENT-VALUE	float	123.2
DESCRIPTION	string	This sensor monitors the core activity.
STATUS-FLAGS	string 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	1010
EVENT-STATE	integer	0
RELIABILITY	integer	0
OUT-OF-SERVICE	boolean	0
NUMBER-OF-STATES	unsigned integer	2
STATE-TEXT	string	test
PRIORITY-ARRAY	string Sequence of **16** values separated by "|". The type of these values can be one of the following (**Null**, **real**, **binary**, **integer**)	NULL|NULL|NULL|NULL|NULL|NULL|12|NULL| NULL|NULL|NULL|NULL|NULL|NULL|NULL|25.000000

	For more information look at ***BACnetPriorityArray*** type on the data types section.	
RELINQUISH-DEFAULT	real	12.3
TIME-DELAY	integer	0
NOTIFICATION-CLASS	integer	0
ALARM-VALUES	string **List of** unsigned integer.	1|1|2
FAULT-VALUES	string **List of** unsigned integer.	1|3
EVENT-ENABLE	string 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	101
ACKED-TRANSITIONS	string 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	101
NOTIFY-TYPE	integer	1
EVENT-TIME-STAMPS	string Sequence of time stamps.	2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0
PROFILE-NAME	string	Test

Schedule Object (SCH)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0
PRESENT-VALUE	float	123.2
DESCRIPTION	string	This sensor monitors the core activity.
EFFECTIVE-PERIOD	string	
WEEKLY-SCHEDULE	string	

EXCEPTION-SCHEDULE	string	
SCHEDULE-DEFAULT	String Bacnet Type Any	2:12 (See Dataype table for more information)
LIST-OF-OBJECT-PROPERTY-REFERENCES	string [OBJECT-IDENTIFIER] | [PROPERTY-IDENTIFIER] | [PROPERTY-ARRAY-INDEX (Optional)] | [DEVICE- IDENTIFIER (Optional)];[OBJECT-IDENTIFIER] | [PROPERTY-IDENTIFIER] | [PROPERTY-ARRAY-INDEX (Optional)] | [DEVICE-IDENTIFIER (Optional)];… [Instance, Object Type] | [Property Identifier] | [Property Array Index] | [Instance, Object Type]; [Instance, Object Type] | [Property Identifier] | [Property Array Index] | [Instance, Object Type];…	This property specifies the Device Identifiers, Object Identifiers and Property Identifiers of the properties to be written with specific values at specific times on specific days. **E.G.: 29,4194303|29|786|30,4194303; 29,4194303|29|786|30,4194303…**
PRIORITY-FOR-WRITING	unsigned integer	
STATUS-FLAGS	string 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	1010
RELIABILITY	integer	0
OUT-OF-SERVICE	boolean	0
PROFILE-NAME	string	Test

Device Object (DEV)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0
SYSTEM-STATUS	integer {operational (0), operational-read-only (1),	4

	download-required (2), download-in-progress (3), non-operational (4), backup-in-progress (5)}	
VENDOR-NAME	string	TEST
VENDOR-IDENTIFIER	integer	12
MODEL-NAME	string	TEST
FIRMWARE-REVISION	string	TEST
APPLICATION-SOFTWARE-VERSION	string	test
LOCATION	string	test
DESCRIPTION	string	This sensor monitors the core activity.
PROTOCOL-VERSION	unsigned integer	4
PROTOCOL-REVISION	unsigned integer	4
PROTOCOL-SERVICES-SUPPORTED	See **BacnetServicesSupported**	
PROTOCOL-OBJECT-TYPES-SUPPORTED	See **BacnetServicesSupported**	
PROFILE-NAME	string	Test

Life Safety Point Object (LSP)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0
PRESENT-VALUE	float	123.2
TRACKING-VALUE	float	(0 – 23) Reflects the non-latched state of the Life Safety Point object.
DESCRIPTION	string	This sensor monitors the core activity
DEVICE-TYPE	string	This is a text description of the physical device that the Life Safety Point object represents.
STATUS-FLAGS	string	1010

	4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	
EVENT-STATE	integer	0
RELIABILITY	integer	0
OUT-OF-SERVICE	boolean	0
MODE	integer	(0 – 14) Shall convey the desired operating mode for the Life Safety Point object.
ACCEPTED-MODES	string	Shall specify all values the Mode property accepts when written to using BACnet services. (10|20|30|40…)
TIME-DELAY	integer	0
NOTIFICATION-CLASS	integer	0
LIFE-SAFETY-ALARM-VALUES	string	This property is required if intrinsic reporting is supported by this object.
ALARM-VALUES	string **List of** unsigned integer.	1|2|3
FAULT-VALUES	string **List of** unsigned integer.	1|2|3
EVENT-ENABLE	string 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	101
ACKED-TRANSITIONS	string 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	101
NOTIFY-TYPE	integer	1
EVENT-TIME-STAMPS	string Sequence of time stamps.	2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0
SILENCED	unsigned integer	(0 – 3) Indicates the silenced state.

OPERATION-EXPECTED	unsigned integer	(0 – 9) Specifies the next operation expected by this object to handle a specific life safety situation.
MAINTENANCE-REQUIRED	unsigned integer	(0 – 3) Indicates the type of maintenance required for the life safety point.
SETTING	unsigned integer	Shall be used to convey the desired setting of the input(s) or process used to determine the logical state of the Present Value.
DIRECT-READING	float	Indicates an analog quantity that reflects the measured or calculated reading from an initiating device.
UNITS	integer	0
MEMBER-OF	string	(0:1,0:2|0:3,0:4|0:5,0:6|...) Shall indicate those Life Safety Zone objects of which this Life Safety Point object is considered to be a zone member.
PROFILE-NAME	string	test

Life Safety Zone Object (LSZ)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0
PRESENT-VALUE	float	123.2
TRACKING-VALUE	float	(0 – 23) Reflects the non-latched state of the Life Safety Point object.
DESCRIPTION	string	This sensor monitors the core activity
DEVICE-TYPE	string	This is a text description of the physical device that the Life Safety Point object represents.
STATUS-FLAGS	string 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	1010

EVENT-STATE	integer	0
RELIABILITY	integer	0
OUT-OF-SERVICE	boolean	0
MODE	integer	(0 – 14) Shall convey the desired operating mode for the Life Safety Point object.
ACCEPTED-MODES	string	Shall specify all values the Mode property accepts when written to using BACnet services. (10|20|30|40…)
TIME-DELAY	integer	0
NOTIFICATION-CLASS	integer	0
LIFE-SAFETY-ALARM-VALUES	string	This property is required if intrinsic reporting is supported by this object.
ALARM-VALUES	string **List of** unsigned integer.	1|2|3
FAULT-VALUES	string **List of** unsigned integer.	1|2|3
EVENT-ENABLE	string 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	101
ACKED-TRANSITIONS	string 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	101
NOTIFY-TYPE	integer	1
EVENT-TIME-STAMPS	string Sequence of time stamps.	2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0
SILENCED	unsigned integer	(0 – 3) Indicates the silenced state.
OPERATION-EXPECTED	unsigned integer	(0 – 9) Specifies the next operation expected by this object to handle a specific life safety situation.
MAINTENANCE-REQUIRED	unsigned integer	(0 – 3) Indicates the type of maintenance required for the life safety point.

ZONE-MEMBERS	string	(0:1,0:2|0:3,0:4|0:5,0:6|...)
MEMBER-OF	string	(0:1,0:2|0:3,0:4|0:5,0:6|...) Shall indicate those Life Safety Zone objects of which this Life Safety Point object is considered to be a zone member.
PROFILE-NAME	string	test

Accumulator Object (ACC)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	1:0
OBJECT-NAME	string	temperatureSensor_core
OBJECT-TYPE	integer	0
PRESENT-VALUE	float	123.2
DESCRIPTION	string	This sensor monitors the core activity
DEVICE-TYPE	string	This property, of type CharacterString, is a text description of the physical device represented by the Accumulator object.
STATUS-FLAGS	string 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	1010
EVENT-STATE	integer	0
RELIABILITY	integer	0
OUT-OF-SERVICE	boolean	0
SCALE	Integer/float/string	(3.14) Indicates the conversion factor to be multiplied with the value of the Present Value property to provide a value in the units indicated by Units.
UNITS	string	0
PRESCALE	string	(-3413;-1119) Presents the coefficients that are used for converting the pulse signals generated by the measuring instrument into the value displayed

		by Present Value.
MAX-PRES-VALUE	integer	0
VALUE-CHANGE-TIME	string	(25-10-2013 15:39:23.27) This read-only property, of type BACnetDateTime, shall be present if the Present_Value property is adjustable by writing to the Value Before Change or Value Set properties.
VALUE-BEFORE-CHANGE	integer	(100) This property, of type Unsigned, indicates the value of the Present_Value property just prior to the most recent write to the Value Set or Value Before Change properties.
VALUE-SET	integer	(786) This property, of type Unsigned, indicates the value of the Present_Value property after the most recent write to the Value Set or Value Before Change properties.
PULSE-RATE	integer	(0) This property, of type Unsigned, shall indicate the number of input pulses received during the most recent period specified by Limit Monitoring Interval.
HIGH-LIMIT	integer	0
LIMIT-MONITORING-INTERVAL	integer	(500) This property, of type Unsigned, specifies the monitoring period in seconds for determining the value of Pulse Rate.
NOTIFICATION-CLASS	integer	0
TIME-DELAY	integer	0
LIMIT-ENABLE	string 2 Boolean values (1 = true and 0 = false) **{lowLimitEnable, highLimitEnable}**	01
EVENT-ENABLE	string 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	101
ACKED-TRANSITIONS	string 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	101
NOTIFY-TYPE	integer	1
EVENT-TIME-STAMPS	string	2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0

	Sequence of time stamps.	
PROFILE-NAME	string	test

Averaging Object (AVR)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	**E.G.: 1:0**
OBJECT-NAME	string	**E.G.: Averaging Object**
OBJECT-TYPE	integer	**E.G.: 18**
MINIMUM-VALUE	float	Shall reflect the lowest value contained within the buffer window for the most recent 'Window Samples' samples. **E.G.: 1.0**
MINIMUM-VALUE-TIMESTAMP	string [Date] | [Time]	Indicates the date and time at which the value stored in Minimum Value was sampled. **E.G.: [1-1-1970] [11:0:0.0]**
AVERAGE-VALUE	float	Shall reflect the average value contained within the buffer window for the most recent 'Window Samples' samples. **E.G.: 1.0**
VARIANCE-VALUE	float	Shall reflect the variance value contained within the buffer window for the most recent 'Window Samples' samples. **E.G.: 1.0**
MAXIMUM-VALUE	float	Shall reflect the highest value contained within the buffer window for the most recent 'Window Samples' samples. **E.G.: 1.0**
MAXIMUM-VALUE-TIMESTAMP	string [Date] | [Time]	Indicates the date and time at which the value stored in Maximum Value was sampled. **E.G.: [1-1-1970] [11:0:0.0]**
DESCRIPTION	string	This sensor monitors the core activity

		E.G.: Averaging Object
ATTEMPTED-SAMPLES	integer	Indicates the number of samples that have been attempted to be collected for the current window. **E.G.: 0**
VALID-SAMPLES	integer	Indicates the number of samples that have been successfully collected for the current window. **E.G.: 0**
OBJECT-PROPERTY-REFERENCE	string [OBJECT-IDENTIFIER] | [PROPERTY-IDENTIFIER] | [PROPERTY-ARRAY-INDEX (Optional)] | [DEVICE- IDENTIFIER (Optional)] [Instance, Object Type] | [Property Identifier] | [Property Array Index] | [Instance, Object Type]	Shall identify the object and property whose value is to be sampled during the 'Window Interval'. **E.G.: 29,4194303|29|786|30,4194303**
WINDOW-INTERVAL	integer	Shall indicate the period of time in seconds over which the minimum, maximum and average values are calculated. **E.G.: 50**
WINDOW-SAMPLES	integer	Shall indicate the number of samples to be taken during the period of time specified by the 'Window Interval' property. **E.G.: 10**
PROFILE-NAME	string	**E.G.: Averaging Profile**

Calendar Object (CAL)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	**E.G.: 1:0**
OBJECT-NAME	string	**E.G.: Calendar Object**
OBJECT-TYPE	integer	**E.G.: 6**
DESCRIPTION	string	This sensor monitors the core activity **E.G.: Calendar Object**

PRESENT-VALUE	boolean	Indicates the current value of the calendar. **E.G.: 1 (TRUE) if the current date is in the Date List and 0 (FALSE) if it is not.**
DATE-LIST	string [DATE]|[DATE-RANGE]|[MONTH/WEEK-OF-MOMTH/DAY-OF-WEEK]…	This property is a List of BACnetCalendarEntry, each of which is either an individual date (Date), range of dates (BACnetDateRange), or month/week-of-month/day-of-week specification (BACnetWeekNDay). **E.G.: 11-2-1981|30-1-2013,11-2-2013|12#4#21…**
PROFILE-NAME	string	**E.G.: Calendar Profile**

Command Object (CMD)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	**E.G.: 1:0**
OBJECT-NAME	string	**E.G.: Object: Command**
OBJECT-TYPE	integer	**E.G.: 6**
DESCRIPTION	string	This sensor monitors the core activity **E.G.: Description: Command**
PRESENT-VALUE	unsigned integer	Indicates which action the Command object is to take or has already taken. **E.G.: 3.**
IN-PROCESS	boolean	Shall be set to TRUE when a value is written to the Present Value property. This **TRUE** value indicates that the Command object has begun processing one of a set of action sequences. Once all of the writes have been attempted by the Command object, the In_Process property shall be set back to **FALSE.** **E.G.: 1/0.**
ALL-WRITES-SUCCESSFUL	boolean	Indicates the success or failure of the sequence of actions that are triggered when the Present Value property is written to. **E.G.: 1/0.**
ACTION	string	This property, of type BACnetARRAY of BACnetActionList, specifies an array of "action lists."

	[Device_Identifier(Optional)],[Object_Identifier], [Property_Identifier], [Property_Array_Index(Optional)], [Property_Value], [Priority(1..16)(Optional)], [Post_Delay(Optional)], [Quit_On_Failure], [Write_Successful]| [Device_Identifier(Optional)],[Object_Identifier], [Property_Identifier], [Property_Array_Index(Optional)], [Property_Value], [Priority(1..16)(Optional)], [Post_Delay(Optional)], [Quit_On_Failure], [Write_Successful]; [Device_Identifier(Optional)],[Object_Identifier], [Property_Identifier], [Property_Array_Index(Optional)], [Property_Value], [Priority(1..16)(Optional)], [Post_Delay(Optional)], [Quit_On_Failure], [Write_Successful]|…	**E.G.: ,5:0,85,,0,,,1,0|8:1,5:2,85,0,5,4,3,0,1;,5:0,85,,0,,,1,0|…**
ACTION-TEXT	string [CharacterString]|[CharacterString]|…	This property, of type BACnetARRAY of CharacterString, shall be used to indicate a text string description for each of the possible values of the Present Value property. **E.G.: Item 0|Item 1|…**
PROFILE-NAME	string	**E.G.: Profile: Command**

Event Enrollment Object (EVT)

Property	Value Format	Example
OBJECT-IDENTIFIER	string [ObjectNumber]:[Instance Number]	**E.G.: 1:0**
OBJECT-NAME	string	**E.G.: Object: Event Enrollment**
OBJECT-TYPE	integer	**E.G.: 9**
DESCRIPTION	string	This sensor monitors the core activity **E.G.: Description: Event Enrollment**
EVENT-TYPE	integer	This read only property, of type BACnetEventType, indicates the type of event algorithm that is to be used to

	[Enumerated - {CHANGE_OF_BITSTRING, CHANGE_OF_STATE, CHANGE_OF_VALUE, COMMAND_FAILURE, FLOATING_LIMIT, OUT_OF_RANGE, BUFFER_READY, CHANGE_OF_LIFE_SAFETY, EXTENDED}].	detect the occurrence of events and report to enrolled devices. **E.G.: 0**
NOTIFY-TYPE	integer	This property, of type BACnetNotifyType, shall convey whether the notifications generated by the monitoring algorithm specified by the Event_Type property should be Events or Alarms. **E.G.: 2**
EVENT-PARAMETERS	string [Choice]|[Time-Delay]|[Bitmask]|[List-Of-Bitstring-Values]	The Event_Parameters property, of type BACnetEventParameter, determines the algorithm used to monitor the referenced object and provides the parameter values needed for this algorithm. **E.G.: COB|10|10101010|10101010;10101010;10101010…**
OBJECT-PROPERTY-REFERENCE	string [OBJECT-IDENTIFIER] | [PROPERTY-IDENTIFIER] | [PROPERTY-ARRAY-INDEX (Optional)] | [DEVICE- IDENTIFIER (Optional)] [Instance, Object Type] | [Property Identifier] | [Property Array Index] | [Instance, Object Type]	This property, of type BACnetDeviceObjectPropertyReference, designates the particular object and property referenced by this Event Enrollment object. The algorithm specified by the Event_Type property is applied to the referenced property in order to determine the Event_State of the event. **E.G.: 29,4194303|29|786|30,4194303**
EVENT-STATE	integer	This property, of type BACnetEventState, contains the current state of the event. **E.G.: 0**
EVENT-ENABLE	string 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	This property, of type BACnetEventTransitionBits, conveys three flags that determine whether notifications are enabled for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL transitions. **E.G.: 101**
ACKED-TRANSITIONS	string 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	This property, of type BACnetEventTransitionBits, shall convey three separate flags that each indicate whether the most recent TO-OFFNORMAL, TO-FAULT, or TO-NORMAL event transitions have been acknowledged, if acknowledgment is required for that transition.

		E.G.: 111
NOTIFICATION-CLASS	integer	This property, of type Unsigned, implicitly references a Notification Class object in the device containing the Event Enrollment object. **E.G.: 0**
EVENT-TIME-STAMPS	string Sequence of time stamps.	This property, of type BACnetARRAY[3] of BACnetTimeStamp, shall convey the times of the last event notifications for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events, respectively. **E.G.: 2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0**
PROFILE-NAME	string	**E.G.: Profile: Event Enrollment**

Loop Object Type (LOP)

Property	Value Format	Example
OBJECT-IDENTIFIER	String [ObjectNumber]:[Instance Number]	**E.G.: 12:0**
OBJECT-NAME	String	**E.G.: Object: Loop**
OBJECT-TYPE	Integer	**E.G.: 12**
PRESENT-VALUE	Float	This property indicates the current output value of the loop algorithm in units of the Output Units property. **E.G.: 100**
DESCRIPTION	String	This sensor monitors the core activity **E.G.: Description: Loop**
STATUS-FLAGS	String 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of the loop. **E.G.: 1101**
EVENT-STATE	String	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. **E.G.: 1.0**

RELIABILITY	Integer	The Reliability property, of type BACnetReliability, provides an indication of whether the Present Value of the loop in question is reliable as far as the BACnet Device or operator can determine and, if not, why. **E.G.: 1**
OUT-OF-SERVICE	Boolean	The Out Of Service property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the algorithm this object represents is or is not in service. **E.G.: 1**
UPDATE-INTERVAL	Integer	This property, of type Unsigned, indicates the interval in milliseconds at which the loop algorithm updates the output (Present Value property). **E.G.: 150**
OUTPUT-UNITS	Integer	This property, of type BACnetEngineeringUnits, indicates the engineering units for the output (Present Value property) of this control loop. **E.G.: 13**
MANIPULATED-VARIABLE-REFERENCE	String [OBJECT-IDENTIFIER] | [PROPERTY-IDENTIFIER] | [PROPERTY-ARRAY-INDEX (Optional)] [Instance, Object Type] | [Property Identifier] | [Property Array Index]	This property is of type BACnetObjectPropertyReference. The output (Present Value) of the control loop is written to the object and property designated by the Manipulated Variable Reference. **E.G.: 29,4194303|29|786**
CONTROLLED-VARIABLE-REFERENCE	String [OBJECT-IDENTIFIER] | [PROPERTY-IDENTIFIER] | [PROPERTY-ARRAY-INDEX (Optional)] [Instance, Object Type] | [Property Identifier] | [Property Array Index]	This property is of type BACnetObjectPropertyReference. The Controlled Variable Reference identifies the property used to set the Controlled Variable Value property of the Loop object. **E.G.: 29,4194303|29|786**
CONTROLLED-VARIABLE-VALUE	Float	This property, of type REAL, is the value of the property of the object referenced by the Controlled Variable Reference property. This control loop compares the Controlled Variable Value with the Setpoint to calculate the error. **E.G.: 0**
CONTROLLED-VARIABLE-UNITS	Integer	This property, of type BACnetEngineeringUnits, indicates the engineering units for the Controlled Variable Value property of this object.

		E.G.: 2
SETPOINT-REFERENCE	String [OBJECT-IDENTIFIER] | [PROPERTY-IDENTIFIER] | [PROPERTY-ARRAY-INDEX (Optional)] | [DEVICE- IDENTIFIER (Optional)] [Instance, Object Type] | [Property Identifier] | [Property Array Index] | [Instance, Object Type]	This property, of type BACnetSetpointReference, is a list of references that has a length of zero or one. A length of zero indicates that the setpoint for this control loop is fixed and is contained in the Setpoint property. **E.G.: 8,0|85|15**
SETPOINT	Float	This property, of type REAL, is the value of the loop setpoint or of the property of the object referenced by the Setpoint Reference, expressed in units of the Controlled Variable Units property. **E.G.: 10**
ACTION	Integer	This property, of type BACnetAction, defines whether the loop is DIRECT or REVERSE acting. **E.G.: 1**
PROPORTIONAL-CONSTANT	Float	This property, of type REAL, is the value of the proportional gain parameter used by the loop algorithm. It may be used to represent any of the various forms of gain for the proportional control mode, such as overall gain, throttling range, or proportional band. **E.G.: 5**
PROPORTIONAL-CONSTANT-UNITS	Integer	This property, of type BACnetEngineeringUnits, indicates the engineering units of the Proportional Constant property of this object. **E.G.: 2**
INTEGRAL-CONSTANT	Float	This property, of type REAL, is the value of the integral gain parameter used by the loop algorithm. **E.G.: 1.2**
INTEGRAL-CONSTANT-UNITS	Integer	This property, of type BACnetEngineeringUnits, indicates the engineering units of the Integral Constant property of this object. **E.G.: 3**
DERIVATIVE-CONSTANT	Float	This property, of type REAL, is the value of the derivative gain parameter used by the loop algorithm. **E.G.: 4.5**
DERIVATIVE-CONSTANT-UNITS	Integer	This property, of type BACnetEngineeringUnits,

		indicates the engineering units of the Derivative Constant property of this object. **E.G.: 7**
BIAS	Float	This property, of type REAL, is the bias value used by the loop algorithm expressed in units of the Output Units property. **E.G.: 1.43**
MAXIMUM-OUTPUT	Float	This property, of type REAL, is the maximum value of the Present Value property as limited by the PID loop algorithm. **E.G.: 100**
MINIMUM-OUTPUT	Float	This property, of type REAL, is the minimum value of the Present Value property as limited by the loop algorithm. **E.G.: 50**
PRIORITY-FOR-WRITING	Integer	Loop objects may be used to control the commandable property of an object. This property, of type Unsigned, provides a priority to be used by the command prioritization mechanism. It identifies the particular priority slot in the Priority Array of the Controlled Variable Reference that is controlled by this loop. It shall have a value in the range 1-16. **E.G.: 27**
COV-INCREMENT	Float	This property, of type REAL, shall specify the minimum change in Present Value that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object. **E.G.: 1.44**
TIME-DELAY	Integer	This property, of type Unsigned, shall specify the minimum period of time in seconds that the difference between the Setpoint and the Controlled Variable Value (the Error) must remain outside the band defined by the Error Limit property before a TO-OFFNORMAL event is generated or within the same band before a TO-NORMAL event is generated. This property is required if intrinsic reporting is supported by this object. **E.G.: 17**
NOTIFICATION-CLASS	Integer	This property, of type Unsigned, shall specify the notification class to be used when handling and generating event notifications for this object. **E.G.: 0**

ERROR-LIMIT	Float	This property, of type REAL, shall convey the absolute magnitude that the difference between the Setpoint and Controlled Variable Value (the Error) must exceed before a TO-OFFNORMAL event is generated. **E.G.: 5**
EVENT-ENABLE	String 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	This property, of type BACnetEventTransitionBits, shall convey three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. This property is required if intrinsic reporting is supported by this object. **E.G.: 101**
ACKED-TRANSITIONS	String 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	This property, of type BACnetEventTransitionBits, shall convey three flags that separately indicate the receipt of acknowledgments for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. **E.G.: 111**
NOTIFY-TYPE	Integer	This property, of type BACnetNotifyType, shall convey whether the notifications generated by the object should be Events or Alarms. This property is required if intrinsic reporting is supported by this object. **E.G.: 21**
EVENT-TIME-STAMPS	String Sequence of time stamps.	This optional property, of type BACnetARRAY[3] of BACnetTimeStamp, shall convey the times of the last event notifications for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events, respectively. **E.G.: 2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0**
PROFILE-NAME	String	**E.G.: Profile: Loop**

Notification Class Object Type (NOT)

Property	Value Format	Example
OBJECT-IDENTIFIER	String [ObjectNumber]:[Instance Number]	**E.G.: 15:0**
OBJECT-NAME	String	**E.G.: Object Name: Notification Class**
OBJECT-TYPE	Integer	**E.G.: 15**
DESCRIPTION	String	This sensor monitors the core activity **E.G.: Description: Notification Class**

NOTIFICATION-CLASS	Integer	This property, of type Unsigned, shall indicate the numeric value of this notification class and shall be equal to the instance number of the Notification Class object. **E.G.: 0**
PRIORITY	String	This property, of type BACnetARRAY[3] of Unsigned, shall convey the priority to be used for event notifications for TOOFFNORMAL, TO-FAULT, and TO-NORMAL events, respectively. **E.G.: 1|NULL|1**
ACK-REQUIRED	String	This property, of type BACnetEventTransitionBits, shall convey three separate flags that represent whether acknowledgment shall be required in notifications generated for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL event transitions, respectively. **E.G.: 110**
RECIPIENT-LIST	String [VALID_DAYS]|[FROM_TIME TO_TIME]|[RECIPIENT (Device or Address)]|[PROCESS_IDENTIFIER]|[ISSUE_CONFIRMED_NOTIFICATIONS]|[TRANSITIONS];...	This property, of type List of BACnetDestination, shall convey a list of one or more recipient destinations to which notifications shall be sent when event-initiating objects using this class detect the occurrence of an event. **E.G.: 1111000|15:21:35.0 0:0:0.0|1,0,88 |0|0|010;1111000|0:0:0.0 0:0:0.0|0,0,0|0|0|111;1111000|0:0:0.0 0:0:0.0|1,1,99 |0|0|111...**
PROFILE-NAME	String	**E.G.: Profile Name: Notification Class**

Program Object Type (PRG)

Property	Value Format	Example
OBJECT-IDENTIFIER	String [ObjectNumber]:[Instance Number]	**E.G.: 16:0**
OBJECT-NAME	String	**E.G.: Object Name: Program Object Type (PRG)**
OBJECT-TYPE	Integer	**E.G.: 16**
PROGRAM-STATE	Integer **0** - IDLE **1** - LOADING **2** - RUNNING **3** - WAITING **4** - HALTED	This property, of type BACnetProgramState, reflects the current logical state of the process executing the application program this object represents. This property is Read-Only. **E.G.: 3**

	5 - UNLOADING	
PROGRAM-CHANGE	Integer **0** - READY **1** - LOAD **2** - RUN **3** - HALT **4** - RESTART **5** - UNLOAD	This property, of type BACnetProgramRequest, is used to request changes to the operating state of the process this object represents. The Program Change property provides one means for changing the operating state of this process. **E.G.: 4**
REASON-FOR-HALT	Integer **0** - NORMAL **1** - LOAD FAILED **2** - INTERNAL **3** - PROGRAM **4** - OTHER	If the process executing the application program this object represents encounters any type of error that causes process execution to be halted, then this property shall reflect the reason why the process was halted. The Reason For Halt property shall be an enumerated type called BACnetProgramError. **E.G.: 1**
DESCRIPTION-OF-HALT	String	This property is a character string that may be used to describe the reason why a program has been halted. **E.G.: InduSoft Halt**
PROGRAM-LOCATION	String	This property is a character string that may be used by the application program to indicate its location within the program code, for example, a line number or program label or section name. The content of this string is a local matter. **E.G.: InduSoft**
DESCRIPTION	String	This property is a string of printable characters that may be used to describe the application being carried out by this process or other locally desired descriptive information. **E.G.: Description: Program Object Type (PRG)**
INSTANCE-OF	String	This property is a character string that is the local name of the application program being executed by this process. The content of this string is a local matter. **E.G.: Program Object Type (PRG) Instance**
STATUS-FLAGS	String 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of the program. **E.G.: 1010**
RELIABILITY	Integer	The Reliability property, of type BACnetReliability, provides an indication of whether the application-specific properties of the program object or the process executing the application program are "reliable" as far as

		the BACnet Device can determine and, if not, why. **E.G.: 9**
OUT-OF-SERVICE	Boolean	The Out Of Service property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the process this object represents is not in service. **E.G.: 1**
PROFILE-NAME	String	**E.G.: Profile Name: Program Object Type (PRG)**

Pulse Converter Object Type (PC)

Property	Value Format	Example
OBJECT-IDENTIFIER	String [ObjectNumber]:[Instance Number]	**E.G.: 24:0**
OBJECT-NAME	String	**E.G.: Object Name: Pulse Converter Object Type (PC)**
OBJECT-TYPE	Integer	**E.G.: 24**
DESCRIPTION	String	**E.G.: Description: Pulse Converter Object Type (PC)**
PRESENT-VALUE	Float	This property, of type REAL, indicates the accumulated value of the input being measured. It is computed by multiplying the current value of the Count property by the value of the Scale Factor property. **E.G.: 100**
INPUT-REFERENCE	String [OBJECT-IDENTIFIER] | [PROPERTY-IDENTIFIER] | [PROPERTY-ARRAY-INDEX (Optional)] [Instance, Object Type] | [Property Identifier] | [Property Array Index]	This optional property, of type BACnetObjectPropertyReference, indicates the object and property (typically an Accumulator object's Present Value property) representing the actual physical input that is to be measured and presented by the Pulse Converter object. **E.G.: 29,4194303|29|786**
STATUS-FLAGS	String 4 Boolean values encoded as string (1 = true and 0 = false) **{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}**	This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a Pulse Converter. **E.G.: 1101**
EVENT-STATE	Integer	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. **E.G.: 1**
RELIABILITY	Integer	The Reliability property, of type BACnetReliability, provides an indication of whether the Present Value and/or Count properties 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. **E.G.: 5**
OUT-OF-SERVICE	Boolean	The Out Of Service property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the input that the object directly represents, if any, is not in service. **E.G.: 1**
UNITS	Integer	This property, of type BACnetEngineeringUnits, indicates the measurement units of the Present Value property. See the BACnetEngineeringUnits ASN.1 production in Clause 21 for a list of engineering units defined by this standard. **E.G.: 7**
SCALE-FACTOR	Float	This property, of type REAL, provides the conversion factor for computing Present Value. It represents the change in Present Value resulting from changing the value of Count by one. **E.G.: 70**
ADJUST-VALUE	Float	This property, of type REAL, is written to adjust the Present Value property (and thus the Count property also) by the amount written to Adjust Value. **E.G.: 10**
COUNT	Integer	This read-only property, of type Unsigned, indicates the count of the input pulses as acquired from the physical input or the property referenced by the Input Reference property. **E.G.: 20**
UPDATE-TIME	String	This read-only property, of type BACnetDateTime, reflects the date and time of the most recent change to the Count property as a result of input pulse accumulation and is updated atomically with the Count property. If no such change has yet occurred, this property shall have wildcard values for all date and time fields. **E.G.: 29-5-2014 17:6:24.78**
COUNT-CHANGE-TIME	String	This read-only property, of type BACnetDateTime, represents the date and time of the most recent

		occurrence of a write to the Adjust Value property. If no such write has yet occurred, this property shall have wildcard values for all date and time fields. **E.G.: 29-5-2014 17:6:24.78**
COUNT-BEFORE-CHANGE	Integer	This property, of type Unsigned, indicates the value of the Count property just prior to the most recent write to the Adjust Value properties. If no such write has yet occurred, this property shall have the value zero. **E.G.: 0**
COV-INCREMENT	Float	This property, of type REAL, shall specify the minimum change in Present Value that will cause a COV notification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object. **E.G.: 30**
NOTIFICATION-CLASS	Integer	This property, of type Unsigned, shall specify the notification class to be used when handling and generating event notifications for this object. **E.G.: 25**
TIME-DELAY	Integer	This property, of type Unsigned, shall specify the minimum period of time in seconds that the Present Value must remain outside the band defined by the High Limit and Low Limit properties before a TO-OFFNORMAL event is generated or remain within the same band, including the Deadband property, before a TO-NORMAL event is generated. This property is required if intrinsic reporting is supported by this object. **E.G.: 80**
HIGH-LIMIT	Float	This property, of type REAL, shall specify a limit that the Present Value must exceed before an event is generated. This property is required if intrinsic reporting is supported by this object. **E.G.: 50**
LOW-LIMIT	Float	This property, of type REAL, shall specify a limit below which the Present Value must fall before an event is generated. This property is required if intrinsic reporting is supported by this object. **E.G.: 60**
DEADBAND	Float	This property, of type REAL, shall specify a range between the High Limit and Low Limit properties, which the Present Value must remain within for a TO-NORMAL event to be generated under these conditions: **a)** the Present Value must fall below the High

		Limit minus Deadband, and **b)** the Present Value must exceed the Low Limit plus the Deadband, and **c)** the Present Value must remain within this range for a minimum period of time, specified in the Time Delay property, and **d)** either the HighLimitEnable or LowLimitEnable flag must be set in the Limit Enable property, and **e)** the TO-NORMAL flag must be set in the Event Enable property This property is required if intrinsic reporting is supported by this object. **E.G.: 40**
LIMIT-ENABLE	String 2 Boolean values (1 = true and 0 = false) **{lowLimitEnable, highLimitEnable}**	This property, of type BACnetLimitEnable, shall convey two flags that separately enable and disable reporting of high limit and low limit offnormal events and their return to normal. This property is required if intrinsic reporting is supported by this object. **E.G.: 10**
EVENT-ENABLE	String 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	This property, of type BACnetEventTransitionBits, shall convey three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. In the context of Pulse Converter objects, transitions to the High Limit or Low Limit Event States are considered to be "offnormal" events. This property is required if intrinsic reporting is supported by this object. **E.G.: 101**
ACKED-TRANSITIONS	String 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	This property, of type BACnetEventTransitionBits, shall convey three flags that separately indicate the receipt of acknowledgements for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. **E.G.: 111**
NOTIFY-TYPE	Integer	This property, of type BACnetNotifyType, shall convey whether the notifications generated by the object should be Events or Alarms. This property is required if intrinsic reporting is supported by this object. **E.G.: 2**
EVENT-TIME-STAMPS	String Sequence of time stamps.	This optional property, of type BACnetARRAY[3] of BACnetTimeStamp, shall convey the times of the last event notifications for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events, respectively. **E.G.: 2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0**

PROFILE-NAME	String	**E.G.: Profile Name: Pulse Converter Object Type (PC)**

Group Object Type (GRP)

Property	Value Format	Example
OBJECT-IDENTIFIER	String [ObjectNumber]:[Instance Number]	**E.G.: 11:0**
OBJECT-NAME	String	**E.G.: Object Name: Group Object Type (GRP)**
OBJECT-TYPE	Integer	**E.G.: 11**
DESCRIPTION	String	**E.G.: Description: Group Object Type (GRP)**
LIST-OF-GROUP-MEMBERS	String [OBJECT-IDENTIFIER]-[LIST-OF-PROPERTY-REFERENCES (Optional)]|… [Instance]:[Object Type]-[{Property Identifier}:{Property Array Index (Optional)},…] (Optional)|…	This property is a list of one or more read access specifications, which defines the members of the group that shall be referenced when this object is specified in a protocol transaction. Each read access specification shall consist of two parts: 1) an Object Identifier and 2) a List Of Property References. **E.G.: 0:0-0:10,0,0|0:0|0:0|0:0-0:10,5:25,0|0:0|0:0-2:30,3:30**
PRESENT-VALUE	String [OBJECT-IDENTIFIER]-[LIST-OF-RESULTS (Optional)]|… [Instance]:[Object Type]-[{Property Identifier}:{([Property Array Index]'[Error Class]'[Error Code]) OR (Property Value)}…] (Optional)|…	This property is a list that contains the values of all the properties specified in the List Of Group Members. This is a "read only" property; it cannot be used to write a set of values to the members of the group. **E.G.: 0:0-0:10'2'50,0:111,0:111|0:0|0:0|0:0-0:10'2'50,5:25'2'32,0:111|0:0|0:0-2:30'2'32,3:30'2'32**
PROFILE-NAME	String	**E.G.: Profile Name: Group Object Type (GRP)**

File Object Type (FO)

Property	Value Format	Example
OBJECT-IDENTIFIER	String [ObjectNumber]:[Instance Number]	**E.G.: 10:0**
OBJECT-NAME	String	**E.G.: Object Name: File Object Type (FO)**
OBJECT-TYPE	Integer	**E.G.: 10**

DESCRIPTION	String	**E.G.: Description: File Object Type (FO)**
FILE-TYPE	String	This property, of type CharacterString, identifies the intended use of this file. **E.G.: JPEG**
FILE-SIZE	Integer	This property, of type Unsigned, indicates the size of the file data in octets. **E.G.: -1**
MODIFICATION-DATE	String	This property, of type BACnetDateTime, indicates the last time this object was modified. A File object shall be considered modified when it is created or written to. **E.G.: 29-5-2014 17:6:24.78**
ARCHIVE	Boolean	This property, of type BOOLEAN, indicates whether the File object has been saved for historical or backup purposes. **E.G.: 1**
READ-ONLY	Boolean	This property, of type BOOLEAN, indicates whether (FALSE) or not (TRUE) the file data may be changed through the use of a BACnet AtomicWriteFile service. **E.G.: 0**
FILE-ACCESS-METHOD	Integer	This property, of type BACnetFileAccessMethod, indicates the type(s) of file access supported for this object. The possible values for File_Access_Method are: {RECORD_ACCESS, STREAM_ACCESS}. **E.G.: 1**
RECORD-COUNT	Integer	This property, of type Unsigned, indicates the size of the file data in records. **E.G.: 0**
PROFILE-NAME	String	**E.G.: Profile Name: File Object Type (FO)**

Trend Log Object Type (TL)

Property	Value Format	Example
OBJECT-IDENTIFIER	String [ObjectNumber]:[Instance Number]	**E.G.: 20:4194302**
OBJECT-NAME	String	**E.G.: Object Name: Trend Log Object Type (TL)**

OBJECT-TYPE	Integer	**E.G.: 20**
DESCRIPTION	String	**E.G.: Description: Trend Log Object Type (TL)**
LOG-ENABLE	Boolean	This property, of type BOOLEAN, indicates and controls whether (TRUE) or not (FALSE) logging is enabled. **E.G.: 1**
START-TIME	String	This property, of type BACnetDateTime, specifies the date and time at or after which logging shall be enabled by this property. **E.G.: 29-5-2014 17:6:24.78**
STOP-TIME	String	This property, of type BACnetDateTime, specifies the date and time at or after which logging shall be disabled by this property. **E.G.: 29-5-2014 17:6:24.78**
LOG-DEVICE-OBJECT-PROPERTY	String [OBJECT-IDENTIFIER] | [PROPERTY-IDENTIFIER] | [PROPERTY-ARRAY-INDEX (Optional)] | [DEVICE- IDENTIFIER (Optional)] [Instance, Object Type] | [Property Identifier] | [Property Array Index] | [Instance, Object Type]	This property, of type BACnetDeviceObjectPropertyReference, specifies the Device Identifier, Object Identifier and Property Identifier of the property to be trend logged. **E.G.: 29,4194303|29|786|30,4194303**
LOG-INTERVAL	Unsigned Integer	This property, of type Unsigned, specifies the periodic interval in hundredths of seconds for which the referenced property is to be logged. **E.G.: 100**
COV-RESUBSCRIPTION-INTERVAL	Unsigned Integer	If the Trend Log is acquiring data from a remote device by COV subscription, this property, of type Unsigned, specifies the number of seconds between COV resubscriptions, provided that COV subscription is in effect. **E.G.: 300**
CLIENT-COV-INCREMENT	Float	If the Trend Log is acquiring COV data, this property, of type BACnetClientCOV, specifies the increment to be used in determining that a change of value has occurred. **E.G.: 1.44**
STOP-WHEN-FULL	Boolean	This property, of type BOOLEAN, specifies whether (TRUE) or not (FALSE) logging should cease when the buffer is full. When logging ceases, Log Enable shall be set FALSE.

		E.G.: 1
BUFFER-SIZE	Unsigned Integer32	This property, of type Unsigned32, shall specify the maximum number of records the buffer may hold. If writable, it may not be written when Log Enable is TRUE. The disposition of existing records when Buffer Size is written is a local matter. **E.G.: 20**
RECORD-COUNT	Unsigned Integer32	This property, of type Unsigned32, shall represent the number of records currently resident in the log buffer. A write of the value zero to this property shall cause all records in the log buffer to be deleted and Records Since Notification to be reset to zero. Upon completion, this event shall be reported in the log as the initial entry. **E.G.: 20**
TOTAL-RECORD-COUNT	Unsigned Integer32	This property, of type Unsigned32, shall represent the total number of records collected by the Trend Log object since creation. **E.G.: 72**
NOTIFICATION-THRESHOLD	Unsigned Integer32	This property, of type Unsigned32, shall specify the value of Records Since Notification at which notification occurs. This property is required if intrinsic reporting is supported by this object. **E.G.: 100**
RECORDS-SINCE-NOTIFICATION	Unsigned Integer32	This property, of type Unsigned32, represents the number of records collected since the previous notification, or since the beginning of logging if no previous notification has occurred. This property is required if intrinsic reporting is supported by this object. **E.G.: 72**
LAST-NOTIFY-RECORD	Unsigned Integer32	This property, of type Unsigned32, represents the SequenceNumber associated with the most recently collected record whose collection triggered a notification. **E.G.: 0**
EVENT-STATE	Integer	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. **E.G.: 0**
NOTIFICATION-CLASS	Unsigned Integer	This property, of type Unsigned, shall specify the notification class to be used when handling and generating event notifications for this object. **E.G.: 0**

EVENT-ENABLE	String 3 Boolean values (1 = true and 0 = false) **{to-offnormal, to-fault, to-normal}**	This property, of type BACnetEventTransitionBits, shall convey three flags that separately enable and disable reporting of TO-FAULT and TO-NORMAL events. **E.G.: 010**
ACKED-TRANSITIONS	String 3 Boolean values (1 = true and 0 = false) **{TO-OFFNORMAL, TO-FAULT, TO-NORMAL}**	This property, of type BACnetEventTransitionBits, shall convey three flags that separately indicate the receipt of acknowledgments for TO-OFFNORMAL, TO-FAULT and TO-NORMAL events. **E.G.: 111**
NOTIFY-TYPE	Integer	This property, of type BACnetNotifyType, shall convey whether the notifications generated by the object should be Events or Alarms. This property is required if intrinsic reporting is supported by this object. **E.G.: 2**
EVENT-TIME-STAMPS	String Sequence of time stamps.	This optional property, of type BACnetARRAY [3] of BACnetTimeStamp, shall convey the times of the last event notifications for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events, respectively. **E.G.: 2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0;2|1-1-1970 11:0:0.0**
PROFILE-NAME	String	**E.G.: Profile Name: Trend Log Object Type (TL)**

The BACNE driver supports the following application services:

Application Service	Initiate Requests
Read-Property-Multiple	●
Read-Property-Single	●
Write-Property-Single	●

Conformance Testing

The following hardware/software was used for conformance testing:

- **Driver Configuration**:
  - **Operation System (development)**: Windows 8
  - **Operation System (target)**: Windows XP + SP2, Windows 7/8 and Windows CE
  - **Studio Version**: v8.0 + P2
  - **Driver Version**: 3.9
  - **Cable**: Ethernet Cable

Driver Version	Studio Version	Operating System (development)	Operating System (runtime)	Equipment
3.11	8.0 + SP1 + P1	Win 7/8	▪ Windows 7/8 ▪ Windows Embedded	▪ SCADA Engine BACnet Simulator ▪ Polarsoft BACNET quick Test

Note:
Once the registers on the BACnet devices are writable and the testing procedures are based on a device simulator which does not supports writing all the addresses we cannot guarantee writing support to all the BACnet objects.

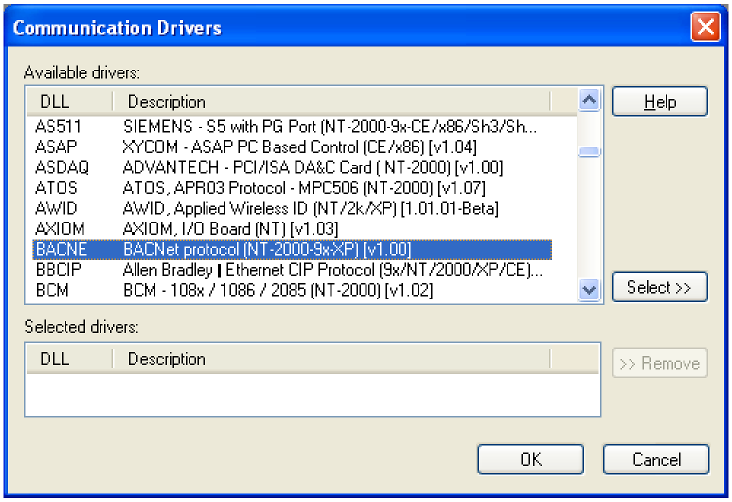
Configuring the Device

Once the selected driver and the target device are both properly configured, it is not necessary to install any other software on your computer to enable communication between the host and the device. All runtime communication is handled within your Studio application project. However, programming the device itself — that is, developing control logic and downloading it to the device — still requires using the device’s own programming tool.

Selecting the Driver

When you install Studio, all of the communication drivers are automatically installed in the **\DRV** subdirectory but they remain dormant until manually selected for specific applications. To select the BACNE driver for your Studio application:

1. From the main menu bar, select **Insert** → **Driver** to open the *Communication Drivers* dialog.
2. Select the **BACNE** driver from the *Available Drivers* list, and then click the **Select** button.



Available drivers:

DLL	Description
AS511	SIEMENS - S5 with PG Port (NT-2000-9x-CE/x86/Sh3/Sh...
ASAP	XYCOM - ASAP PC Based Control (CE/x86) [v1.04]
ASDAQ	ADVANTECH - PCI/ISA DA&C Card (NT-2000) [v1.00]
ATOS	ATOS, APR03 Protocol - MPC506 (NT-2000) [v1.07]
AWID	AWID, Applied Wireless ID (NT/2k/XP) [1.01.01-Beta]
AXIOM	AXIOM, I/O Board (NT) [v1.03]
BACNE	BACNet protocol (NT-2000-9x-XP) [v1.00]
BBCIP	Allen Bradley I Ethernet CIP Protocol (9x/NT/2000/XP/CE)...
BCM	BCM - 108x / 1086 / 2085 (NT-2000) [v1.02]

Selected drivers:

DLL	Description

Communication Drivers Dialog

3. When the **BACNE** driver is displayed in the **Selected Drivers** list, click the **OK** button to close the dialog. The driver is added to the *Drivers* folder, in the *Comm* tab of the Workspace.

Note:
It is not necessary to install any other software on your computer to enable communication between Studio and your target device. However, this communication can only be used by the Studio application; it cannot be used to download control logic to the device. For more information, please consult the documentation provided by the device manufacturer.

Attention:
For safety reasons, you must take special precautions when installing any physical hardware. Please consult the manufacturer’s documentation for specific instructions.

Configuring the Driver

Once you have selected the BACNE driver in Studio, you must properly configure it to communicate with your target device. First, you must set the driver's communication settings to match the parameters set on the device. Then, you must build driver worksheets to associate database tags in your Studio application with the appropriate addresses (registers) on the device.

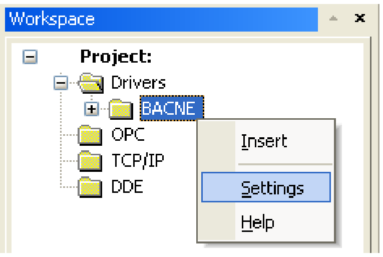
Configuring the Communication Settings

The communication settings are described in detail in the "Communication" chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only BACNE driver-specific settings and procedures will be discussed here. To configure the communication settings for the BACNE driver:

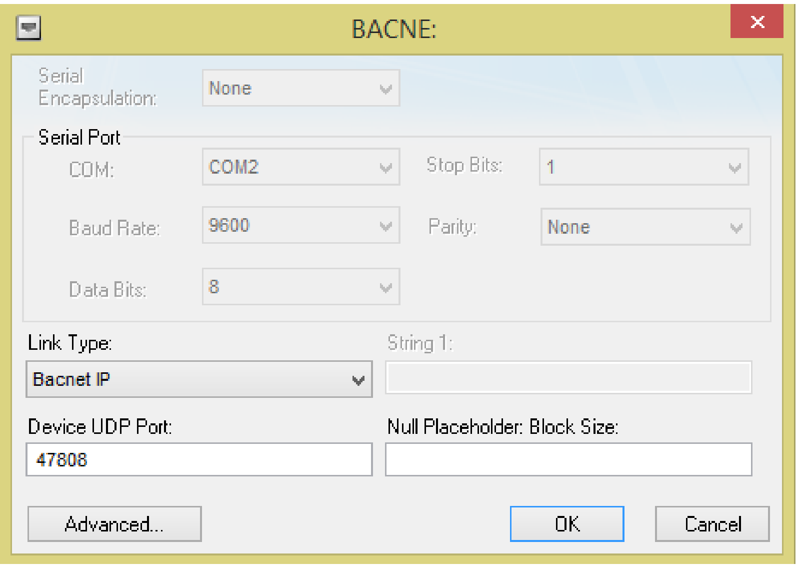
1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The BACNE driver is listed here as a subfolder.

2. Right-click on the *BACNE* subfolder and then select the **Settings** option from the pop-up menu:



Select Settings from the Pop-Up Menu

The *BACNE: Communication Settings* dialog is displayed:



Serial Encapsulation: None

Serial Port

COM: COM2

Stop Bits: 1

Baud Rate: 9600

Parity: None

Data Bits: 8

Link Type: Bacnet IP

String 1:

Device UDP Port: 47808

Null Placeholder: Block Size:

BACNE: Communication Settings Dialog

3. In the *Communication Settings* dialog, configure the driver settings to enable communication with your target device. To ensure error-free communication, the driver settings must *exactly match* the corresponding settings on the device. Please consult the manufacturer's documentation for instructions how to configure the device and for complete descriptions of the settings.

   Depending on your circumstances, you may need to configure the driver *before* you have configured your target device. If this is the case, then take note of the driver settings and have them ready when you later configure the device.

   ➲ **Attention:**
   For safety reasons, you **must** take special precautions when connecting and configuring new equipment. Please consult the manufacturer's documentation for specific instructions.

   The communication settings and their possible values are described in the following table:

Parameters	Default Values	Valid Values	Description
Device UDP port	**47808**	**1** to **65535**	Specify the UDP port. If there is nothing configured, then the driver will use the default value of **47808(BAC0h)**.
Link Type	**Bacnet IP**	**Bacnet IP**	Specify the interface to use.
Null Placeholder: Block Size	**X**	For **Null Placeholder:** A String with the character to be use as placeholder For **Block Size**: A integer number	The **Null Placeholder** is used for writing operation and just for simple datatypes (the ones that do not have specific formats) like present-values, it is a way to send the Bacnet Null Value. For example if for the user the value -333 means NULL specify the NullPlaceholder as -333. The **Block Size** is the number of objects that can be requested on each message.

   You do not need to change any other advanced settings at this time. You can consult the Studio *Technical Reference Manual* later for more information about configuring these settings.

4. Click **OK** to close the *Communication Settings* dialog.

Configuring the Driver Station

To configure the station follow this pattern:

<SubNet>:<Device ID>

Where:

<SubNet> is the BACnet virtual network number,

<Device ID> is the BACnet device ID number or the BACnet device instance

Examples:

Stations
1111:12
1234:32
4444:2
12:2
1000:2

Other available formats:

> ➲ **Attention:**
> You cannot leave the **Station** and **Header** fields blank.

<BacNet IP Address> or

<BacNet IP Address>:<SubNet>:<Device ID>

Where:

<BacNet IP Address> is the BACnet device IP Address,

<SubNet> is the BACnet virtual network number,

<Device ID> is the BACnet Device Instance or the BACnet device ID number.

Examples:

Stations
192.168.110.101
192.168.110.101:1234:32
192.168.110.101:1111:12

Other available format:

<BacNet IP Address>:<SubNet>:0:<BACnet MAC Address>

Where:

<BacNet IP Address> is the BACnet device IP Address,

<SubNet> is the BACnet virtual network number,

<Device ID> or Device Instance is not used in this syntax and should be '0'.

`Note: Any value used as the third parameter in station field will be ignored if there is a fourth parameter of BACnet MAC Address defined.`

<BACnet MAC Address> is a hex string which is the BACnet MAC address of the device.

Example: 192.168.110.101:1111:0:7F

✍ **Note when using the *MAC Address (Technical Details) in the syntax '<BacNet IP Address>:<SubNet>:0:<BACnet MAC Address>'***

The length of the hex string used as BACnet MAC address parameter will set the Destination MAC Layer Address Length in the protocol where every 2 characters in the string is of length 1 byte.

For example:

BACnet MAC Address in Decimal	BACnet MAC Address (hex)	Station	Destination MAC Layer Address Length that will be set in the protocol
1	010000000000	192.168.110.101:1111:0:010000000000	6 Bytes
127	7F	192.168.110.101:1111:0:7F	1 Byte
127	7F0000	192.168.110.101:1111:0:7F0000	3 Bytes
127	7F00000000	192.168.110.101:1111:0:7F00000000	5 Bytes
127	7F0000000000	192.168.110.101:1111:0:7F0000000000	6 Bytes
300	12C0	192.168.110.101:1111:0:12C0	2 Bytes

Refer to your device documentation for the Destination MAC Layer Address Length supported by the device.

Configuring the Driver Worksheets

Each selected driver includes a Main Driver Sheet and one or more Standard Driver Worksheets. The Main Driver Sheet is used to define tag/register associations and driver parameters that are in effect at all times, regardless of application behavior. In contrast, Standard Driver Worksheets can be inserted to define additional tag/register associations that are triggered by specific application behaviors.

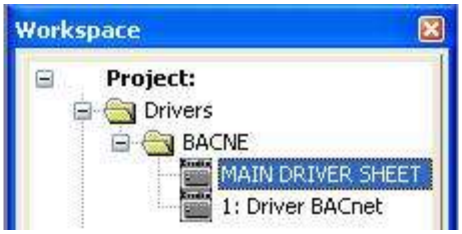
The configuration of these worksheets is described in detail in the "Communication" chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only BACNE driver-specific parameters and procedures are discussed here.

MAIN DRIVER SHEET

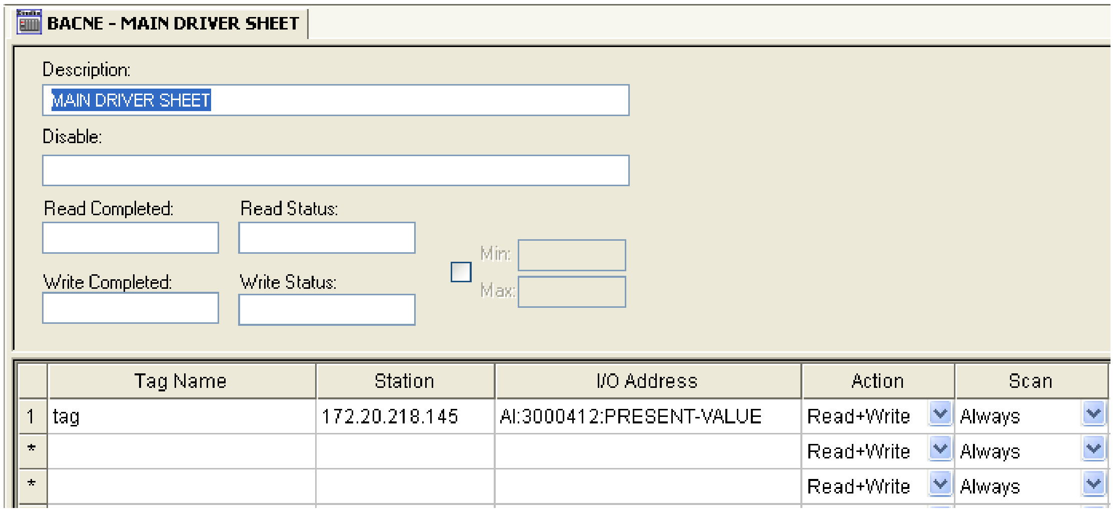
When you select the BACNE driver and add it to your application, Studio automatically inserts the *Main Driver Sheet* in the *BACNE* driver subfolder. To configure the Main Driver Sheet:

1. Select the *Comm* tab in the *Workspace* pane.
2. Open the *Drivers* folder, and then open the *BACNE* subfolder:



Main Driver Sheet in the BACNE Subfolder

3. Double-click on the **MAIN DRIVER SHEET** icon to open the following worksheet:



BACNE - MAIN DRIVER SHEET

Description: MAIN DRIVER SHEET

Disable:

Read Completed:

Read Status:

Write Completed:

Write Status:

□ Min: Max:

	Tag Name	Station	I/O Address	Action	Scan
1	tag	172.20.218.145	AI:3000412:PRESENT-VALUE	Read+Write	Always
*				Read+Write	Always
*				Read+Write	Always

Opening the Main Driver Sheet

Most of the fields on this sheet are standard for all drivers; see the "Communication" chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station** and **I/O Address** fields use syntax that is specific to the BACNE driver.

4. For each table row (i.e., each tag/register association), configure and **I/O Address** fields as follows:

- **I/O Address:** Specify the address of the associated device register, using the following syntax:

  <Type>*:*<Instance>*:*<Property>

  Example — **AI:0:OBJECT-NAME**

  <Type>*:*<Instance>*:*<Property>:<Priority Array> (Only for write)

  Example — **AI:0:OBJECT-NAME:4**

Where**:**

– ***<Type>*** is the register type. Valid values are **AI, AO, AV, BI, BO, BV, DEV, MSI, MSO, MSV, SCH, LSP, LSZ, ACC, AVR, CAL, CMD, EVT, LOP, NOT, PRG, PC, GRP, FO and TL.**

– ***<Instance>*** is the instance of the given register type. If you do not specify this parameter, then Studio will insert a default value of **0**.

– ***<Property>*** is the specific property of the instance. For a list of valid properties, see the table on page 4. If you do not specify this parameter, then Studio will insert a default value of **OBJECT-IDENTIFIER**.

– ***<Priority>*** is used only for writing, this is the priority for writing, and the default value is 16.

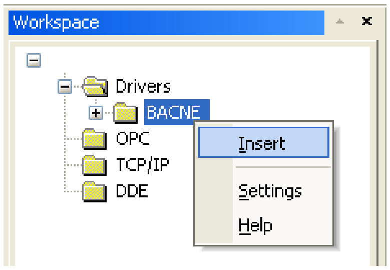
STANDARD DRIVER SHEET

When you select the BACNE driver and add it to your application, it has only a Main Driver Sheet by default (see previous section). However, you may insert additional Standard Driver Worksheets to define tag/register associations that are triggered by specific application behaviors. Doing this will optimize communication and improve system performance by ensuring that tags/registers are scanned only when necessary – that is, only when the application is performing an action that requires reading or writing to those specific tags/registers.

> ✍ **Note:**
> We recommend configuring device registers in sequential blocks in order to maximize performance.

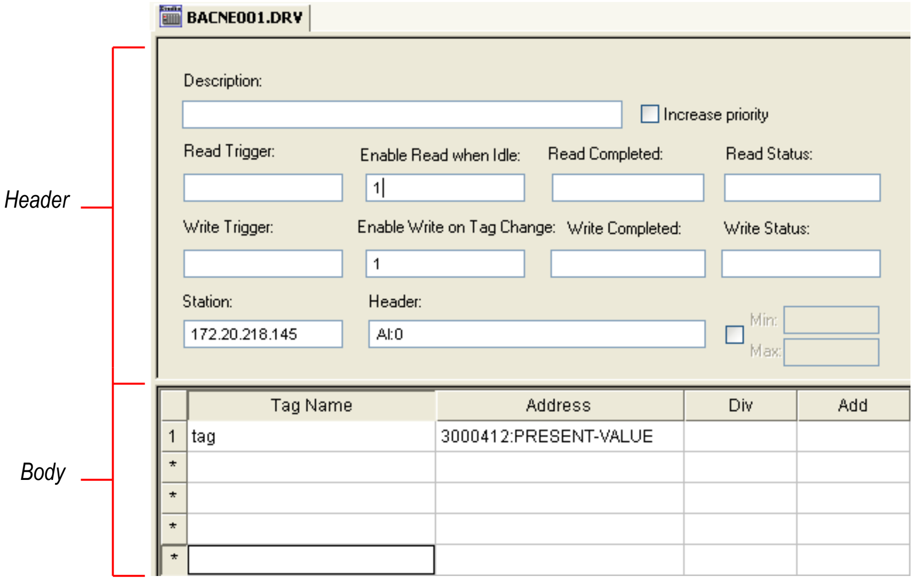
To insert a new Standard Driver Worksheet:

1. In the *Comm* tab, open the *Drivers* folder and locate the *BACNE* subfolder.
2. Right-click on the *BACNE* subfolder, and then select **Insert** from the pop-up menu:



Inserting a New Worksheet

A new BACNE driver worksheet is inserted into the *BACNE* subfolder, and the worksheet is opened for configuration:



Description:

□ Increase priority

Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:
	1		

Write Trigger:	Enable Write on Tag Change:	Write Completed:	Write Status:
	1		

Station: 172.20.218.145

Header: AI:0

□ Min: Max:

	Tag Name	Address	Div	Add
1	tag	3000412:PRESENT-VALUE		
*				
*				
*				
*				

BACNE Driver Worksheet

> **Note:**
>
> Worksheets are numbered in order of creation, so the first worksheet is **`BACNE001.drv`**.

Most of the fields on this worksheet are standard for all drivers; see the "Communication" chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station**, **Header**, and **Address** fields use syntax that is specific to the BACNE driver.

3. Configure the **Header** fields as follows:

   - **Header** field: Specify the address of the first instance in a block of instances on the target device and the type of variable to be read from or write to the device. The addresses declared in the *Body* of the worksheet are simply offsets of this **Header** address. When Read/Write operations are executed for the entire worksheet (see **Read Trigger** and **Write Trigger** above), it scans the entire block of instances from the first address to the last.

     The **Header** field uses the following syntax:

     `<Type>:<Instance Reference>`

     Example — **`AI:5`**

     Where**:**

     – ***`<Type>`*** is the register type. Valid values are **`AI, AO, AV, BI, BO, BV, DEV, MSI, MSO, MSV, SCH, LSP, LSZ, ACC, AVR, CAL, CMD, EVT, LOP, NOT, PRG, PC, GRP, FO and TL`**; and

– ***<Instance Reference>*** is the initial instance (reference) of the block of instances configured on this worksheet.

After you edit the **Header** field, Studio checks the syntax to determine if it is valid. If no value is entered, then the default value is **AI:0**.

You can also specify an indirect tag (e.g. **{header}**), but the tag that is referenced must follow the same syntax and contain a valid value.

➲ **Attention:**
You cannot leave the **Station** and **Header** fields blank.

4. For each table row (i.e., each tag/register association), configure the **Address** field using the following syntax…

   <Instance Offset>*:*<Property>

   Examples — **0:OBJECT-NAME**, **1:PRESENT-VALUE**, **2:LOW-LIMIT**, or **0:77**, **1:85**, **2:59**

   Where:

   – ***<Instance Offset>*** is the value added to ***<Instance Reference>*** parameter (configured in the **Header** field above) to produce the complete instance number; and

   – ***<Property>*** is the specific property of the instance. For a list of valid properties, please see the table on page 4.

For examples of how device registers are specified using **Header** and **Address**, see the following table:

Address on the Device	Header Field	Address Field
AI:0:OBJECT-NAME	**AI:0**	**0:OBJECT-NAME**
AI:1:OBJECT-NAME	**AI:0**	**1:OBJECT-NAME**
AI:1:OBJECT-NAME	**AI:1**	**0:OBJECT-NAME**
AI:2:OBJECT-NAME	**AI:1**	**1:OBJECT-NAME**
AI:5:OBJECT-NAME	**AI:3**	**2:OBJECT-NAME**

For more information about registers and addressing, please consult the manufacturer's documentation.

DATA TYPES

The BACnet protocol uses several data types to transport values from devices. The primitive data types, such as integers, strings and floating-point numbers are easily used and understood. However, some properties employ enumerations, dates and times to represent its data.

Enumerations

The enumerations are associated with textual values on the properties that use this data type. The BACNE driver returns the numeric value of these enumerations on the tags used to read them. The following enumerations are currently used, as seen on the table of page 4:

Enumeration	Values
BACnetEventState	normal (0), fault (1), offnormal (2), high-limit (3), low-limit (4), life-safety-alarm (5)
BACnetNotifyType	alarm (0), event (1), ack-notification (2)
BACnetPolarity	normal (0), reverse (1)
BACnetBinaryPV	inactive (0), active (1)
Tag Number	NULL (0), Boolean (1), Unsigned Integer (2), Integer (3), Real (4), String (7), Enumerated (9)

The BACnetEngineeringUnits enumeration is also used, on the UNITS property. However, due to space limitations, its' possible values are not shown here.

Dates

On the BACNE driver, single dates are **always** represented as a 3-part string with the following format:

DD-MM-YYYY

Where:

DD – is the number of the day (1 to 31)

MM – is the month number (1 to 12)

YYYY – is the year (1900 to 2154)

Dates might be composed to form ranges, and/or with times form a specific moment or range.

Using values outside of the ranges specified here will produce unspecified behavior.

Times

On the BACNE driver, single times are always represented as a 4-part string with the following format:

HH:MM:SS.HS

Where:

HH – is the hour (0 to 23)

MM – is the minute (0 to 59)

SS – is the second (0 to 59)

HS – is the hundredth of second (0 to 99)

SCHEDULE OBJECT

The schedule object has special properties that demand further explanations on its syntax. The effective period, weekly schedule and exception schedule are these properties, combining dates and times on ranges and periods.

Effective Period:

The tag must have a value complying to the following syntax:

```
<start date>|<end date>
```

Where:

<date> is a date as specified before, on the data types section. As the effective period property implies a period, two dates must be supplied, separated by the **|** (pipe) character.

For example a period starting at Apr/12/2010 and ending at May/14/2010 shall be:

```
12-4-2010|14-5-2010
```

> ✍ **Note:**
>
> When reading values if no period or if a field of the date is unspecified, the field is filled with the character **x**. For example, if the entire date is unspecified:
>
> **xx-xx-xxxx|xx-xx-xxxx**

Weeekly Schedule:

The weekly schedule is used to specify times and values for specific days. This property combines times with enumerations for types and values, onto an array of 7 elements (corresponding to the 7 days of a week). Each element of the array may also have multiple times. Starting with a single time instance, the following syntax is used:

`<time instance> = <time>,<type>,<value>`

Where:

`<time>` is a time as specified on data types section

`<type>` is one of the elements of the Tag Number enumeration

`<value>` is the value of **`<type>`** associated with this time

A single day may contain **zero or more** time instances. Those are divided by a semi-colon character, forming a daily schedule element:

`<daily schedule> = [<time instance>;<time instance>;...]`

This syntax means that each day might have any number of time configurations on the schedule, using a semi-colon to separate them. The following example clarifies this syntax:

- Time 12h 25m 32s 500ms	Type: Boolean (1)	Value: True (1)
- Time 13h 40m 3s 20ms	Type: Boolean (1)	Value: False (0)

Value for the day: **`12:25:32.50,1,1;13:40:3.2,1,0`**

It is important to notice that the above example was only for one day but it is necessary to specify the seven days even though some of them may not be defined.

✍ **Note:**

Separate the days using the “|” (pipe) character

So, the complete syntax of a weekly-schedule is:

`<monday>|<tuesday>|<wednesday>|<thursday>|<friday>|<saturday>|<sunday>`

Each of the days of the week is a **`<daily schedule>`**, which means it does not need to have a value.

Example:

The complete and valid value for the weekly-schedule:

```
12:34:56.99,4,5.12;22:10:0.0,2,135|||4:5:12.0,1,1|||1:2:3.4,7,driver BACNE
```

In this example we have the seven days, with schedules for Monday, Thursday and Sunday, summarized on the table:

Day of Week	Time of Day	Type	Value
Monday	12:34:56.99	4 (Real)	5.12
Monday	22:10:0.0	2 (Uint)	135
Thursday	4:5:12.0	1 (Boolean)	TRUE
Sunday	1:2:3.4	7 (String)	driver BACNE

The value configured on the schedule will be read on the **PRESENT-VALUE** address of the object on its time of activity.

Exception Scheduler:

The Exception Scheduler values are built with a sequence of groups. Each group is composed of a date, a list of time and values and a priority value. The list of time_value are separated by a comma (,), whereas the parts of the group are separated by a semi-colon (;) character:

```
<group> = <date>;[<time_value>,< time_value >,...];<priority>
```

Notice that **<date>** syntax is specified on the data types section and the priority is an unsigned integer number.

The **<time_value>** is composed of a time and a value separated by (/) . The value is composed of a datatype number (see datatype table) and a value separated by (:).

Example of a time 10:15:30.50.

Example of a value like unsigned integer 10 will be 2:10 (2 is the datatype number of the unsigned integer)

Now the time_value for this will be 10:15:30.50/2:10

The first group differs from the others only by the date part:

```
<first group> = <start date>,<end date>;[<time_value >,<time_value>,...];<priority>
```

The complete syntax of the exception scheduler is composed of the first group and zero or more groups, divided by the **|** (pipe) character:

```
<first group>[|<group>|<group>|<group>...]
```

Example of an exception scheduler with one group:

26-02-2013;12:1:2.0/2:10;0

Example of a more complex exception scheduler with two groups:

26-2-2013;12:1:2.0/2:10;0|0-0-1900;0:0:0.0/0:NULL,0:0:0.0/1:0,0:0:0.0/7:BacneTest;0

Example of an exception scheduler with starting & ending date with one group:

16-11-2013,29-11-2013;3:10:0.0/1:1,11:0:0.0/1:0,14:50:0.0/1:1,23:59:59.0/1:0;2|

Property Block Size Table:

Per request the maximum APDU supported by BACnet IP is **1476** octets. The following is the table which shows the maximum length of the each property when requested.

Property	Value Format	Maximum Length (In Bytes)	Comments
OUT-OF-SERVICE	Boolean	10	
IN-PROCESS	Boolean	10	
ALL-WRITES-SUCCESSFUL	Boolean	10	
LOG-ENABLE	Boolean	10	
STOP-WHEN-FULL	Boolean	10	
DAYLIGHT-SAVINGS-STATUS	Boolean	10	
ARCHIVE	Boolean	10	
READ-ONLY	Boolean	10	
OBJECT-TYPE	Integer	11	
EVENT-STATE	Integer	11	
RELIABILITY	Integer	11	
UNITS	Integer	11	
TIME-DELAY	Integer	11	
NOTIFICATION-CLASS	Integer	11	
NOTIFY-TYPE	Integer	11	
UPDATE-INTERVAL	Integer	11	
PRIORITY-FOR-WRITING	Integer	11	
ACTION	Integer	11	
OUTPUT-UNITS	Integer	11	
CONTROLLED-VARIABLE-UNITS	Integer	11	
PROPORTIONAL-CONSTANT-UNITS	Integer	11	
INTEGRAL-CONSTANT-UNITS	Integer	11	
DERIVATIVE-CONSTANT-UNITS	Integer	11	
ATTEMPTED-SAMPLES	Integer	11	
VALID-SAMPLES	Integer	11	
WINDOW-INTERVAL	Integer	11	
WINDOW-SAMPLES	Integer	11	
TRACKING-VALUE	Integer	11	
MODE	Integer	11	
SILENCED	Integer	11	
OPERATION-EXPECTED	Integer	11	
MAINTENANCE-REQUIRED	Integer	11	
NUMBER-OF-STATES	Integer	11	
SETTING	Integer	11	
RECORD-COUNT	Integer	11	
LOG-INTERVAL	Integer	11	

BUFFER-SIZE	Integer	11	
TOTAL-RECORD-COUNT	Integer	11	
NOTIFICATION-THRESHOLD	Integer	11	
RECORDS-SINCE-NOTIFICATION	Integer	11	
LAST-NOTIFY-RECORD	Integer	11	
COUNT	Integer	11	
COUNT-BEFORE-CHANGE	Integer	11	
COV-PERIOD	Integer	11	
EVENT-TYPE	Integer	11	
SYSTEM-STATUS	Integer	11	
VENDOR-IDENTIFIER	Integer	11	
DATABASE-REVISION	Integer	11	
PROTOCOL-VERSION	Integer	11	
PROTOCOL-REVISION	Integer	11	
SEGMENTATION-SUPPORTED	Integer	11	
MAX-SEGMENTS-ACCEPTED	Integer	11	
NUMBER-OF-APDU-RETRIES	Integer	11	
MAX-MASTER	Integer	11	
MAX-INFO-FRAMES	Integer	11	
BACKUP-FAILURE-TIMEOUT	Integer	11	
ALARM-VALUE	Integer	11	
MINIMUM-OFF-TIME	Integer	11	
MINIMUM-ON-TIME	Integer	11	
POLARITY	Integer	11	
FEEDBACK-VALUE	Integer	11	
FILE-ACCESS-METHOD	Integer	11	
VALUE-BEFORE-CHANGE	Integer	11	
VALUE-SET	Integer	11	
PULSE-RATE	Integer	11	
LIMIT-MONITORING-INTERVAL	Integer	11	
PROGRAM-STATE	Integer	11	
PROGRAM-CHANGE	Integer	11	
REASON-FOR-HALT	Integer	11	
COV-RESUBSCRIPTION-INTERVAL	Integer	12	
MAX-APDU-LENGTH-ACCEPTED	Integer	12	
APDU-SEGMENT-TIMEOUT	Integer	12	
APDU-TIMEOUT	Integer	12	
CHANGE-OF-STATE-COUNT	Integer	12	
ELAPSED-ACTIVE-TIME	Integer	12	
UTC-OFFSET	Integer	12	
FILE-SIZE	Integer	14	
PRESENT-VALUE	Real	14	
HIGH-LIMIT	Real	14	
LOW-LIMIT	Real	14	
RELINQUISH-DEFAULT	Real	14	
MAX-PRES-VALUE	Real	14	

COV-INCREMENT	Real	14	
DEADBAND	Real	14	
CONTROLLED-VARIABLE-VALUE	Real	14	
SETPOINT	Real	14	
PROPORTIONAL-CONSTANT	Real	14	
INTEGRAL-CONSTANT	Real	14	
DERIVATIVE-CONSTANT	Real	14	
BIAS	Real	14	
MAXIMUM-OUTPUT	Real	14	
MINIMUM-OUTPUT	Real	14	
ERROR-LIMIT	Real	14	
MINIMUM-VALUE	Real	14	
AVERAGE-VALUE	Real	14	
VARIANCE-VALUE	Real	14	
MAXIMUM-VALUE	Real	14	
MIN-PRES-VALUE	Real	14	
RESOLUTION	Real	14	
DIRECT-READING	Real	14	
CLIENT-COV-INCREMENT	Real	14	
SCALE-FACTOR	Real	14	
ADJUST-VALUE	Real	14	
OBJECT-IDENTIFIER	String	14	
OBJECT-NAME	String	9 + (Size of Object Name in characters)	
DESCRIPTION	String	9 + (Size of Description in characters)	
STATUS-FLAGS	String	12	
LIMIT-ENABLE	String	12	
EVENT-ENABLE	String	12	
ACKED-TRANSITIONS	String	11	
PROFILE-NAME	String	9 + (Size of Profile Name in characters)	
PRIORITY-ARRAY	String	NA	
EVENT-TIME-STAMPS	String	NA	
MANIPULATED-VARIABLE-REFERENCE	String	18	
CONTROLLED-VARIABLE-REFERENCE	String	18	
SETPOINT-REFERENCE	String	18	
ACTION (COMMAND)	String	11+ **Per count in the Action** **Per item in the per Action count** 28	
ACTION-TEXT	String	9 + ([3 + Size of Action Text in characters] + [3 + Size of Action Text in characters] + [3 + Size of Action Text in characters]…)	

<table>
<tr><td>MINIMUM-VALUE-TIMESTAMP</td><td>String</td><td colspan="3">19</td><td></td></tr>
<tr><td>MAXIMUM-VALUE-TIMESTAMP</td><td>String</td><td colspan="3">19</td><td></td></tr>
<tr><td>OBJECT-PROPERTY-REFERENCE</td><td>String</td><td colspan="3">23</td><td></td></tr>
<tr><td>LIST-OF-GROUP-MEMBERS</td><td>String</td><td colspan="3">NA</td><td></td></tr>
<tr><td>DEVICE-TYPE</td><td>String</td><td colspan="3">12 + (Size of Device Type in characters)</td><td></td></tr>
<tr><td>ALARM-VALUES</td><td>String</td><td colspan="3">9 + (2 * Number of Alarm Values count)</td><td></td></tr>
<tr><td>FAULT-VALUES</td><td>String</td><td colspan="3">9 + (2 * Number of Fault Values count)</td><td></td></tr>
<tr><td>ACCEPTED-MODES</td><td>String</td><td colspan="3">NA</td><td></td></tr>
<tr><td>LIFE-SAFETY-ALARM-VALUES</td><td>String</td><td colspan="3">9 + (2 * Number of Life Safety Alarm Values count)</td><td></td></tr>
<tr><td>MEMBER-OF</td><td>String</td><td colspan="3">9 + (10 * Number of Member Of count)</td><td></td></tr>
<tr><td>ZONE-MEMBERS</td><td>String</td><td colspan="3">NA</td><td></td></tr>
<tr><td>STATE-TEXT</td><td>String</td><td colspan="3">9 + ([3 + Size of State Text in characters] + [3 + Size of State Text in characters] + [3 + Size of State Text in characters]…)</td><td></td></tr>
<tr><td>LOG-DEVICE-OBJECT-PROPERTY</td><td>String</td><td colspan="3">26</td><td></td></tr>
<tr><td>START-TIME</td><td>String</td><td colspan="3">19</td><td></td></tr>
<tr><td>STOP-TIME</td><td>String</td><td colspan="3">19</td><td></td></tr>
<tr><td>LOG-BUFFER</td><td>String</td><td colspan="3">NA</td><td></td></tr>
<tr><td>INPUT-REFERENCE</td><td>String</td><td colspan="3">19</td><td></td></tr>
<tr><td>UPDATE-TIME</td><td>String</td><td colspan="3">19</td><td></td></tr>
<tr><td>COUNT-CHANGE-TIME</td><td>String</td><td colspan="3">19</td><td></td></tr>
<tr><td>PRIORITY</td><td>String</td><td colspan="3">15</td><td></td></tr>
<tr><td>ACK-REQUIRED</td><td>String</td><td colspan="3">12</td><td></td></tr>
<tr><td>RECIPIENT-LIST</td><td>String</td><td colspan="3">9 + (24 * Number of Recipient List count)</td><td></td></tr>
<tr><td rowspan="9">EVENT-PARAMETERS</td><td rowspan="9">String</td><td rowspan="9">9+</td><td><b>Event Type</b></td><td><b>Size</b></td><td rowspan="9"></td></tr>
<tr><td>CHANGE OF BITSTRING</td><td>NA</td></tr>
<tr><td>CHANGE OF STATE</td><td>NA</td></tr>
<tr><td>CHANGE OF VALUE</td><td>NA</td></tr>
<tr><td>COMMAND FAILURE</td><td>21</td></tr>
<tr><td>FLOATING LIMIT</td><td>36</td></tr>
<tr><td>OUT OF RANGE</td><td>19</td></tr>
<tr><td>BUFFER READY</td><td>8</td></tr>
<tr><td>CHANGE OF LIFE SAFETY</td><td>NA</td></tr>
<tr><td>VENDOR-NAME</td><td>String</td><td colspan="3">12 + (Size of Vendor Name in characters)</td><td></td></tr>
<tr><td>MODEL-NAME</td><td>String</td><td colspan="3">12 + (Size of Model Name in characters)</td><td></td></tr>
<tr><td>FIRMWARE-REVISION</td><td>String</td><td colspan="3">12 + (Size of Firmware Revision in characters)</td><td></td></tr>
<tr><td>APPLICATION-SOFTWARE-VERSION</td><td>String</td><td colspan="3">12 + (Size of Application Software Version in characters)</td><td></td></tr>
<tr><td>LOCATION</td><td>String</td><td colspan="3">12 + (Size of Location in characters)</td><td></td></tr>
<tr><td>OBJECT-LIST</td><td>String</td><td colspan="3">9 + (5 * Number of Object List count)</td><td></td></tr>
<tr><td>PROTOCOL-SERVICES-SUPPORTED</td><td>String</td><td colspan="3">17</td><td></td></tr>
<tr><td>PROTOCOL-OBJECT-TYPES-SUPPORTED</td><td>String</td><td colspan="3">16</td><td></td></tr>
<tr><td>VT-CLASSES-SUPPORTED</td><td>String</td><td colspan="3">NA</td><td></td></tr>
<tr><td>ACTIVE-VT-SESSIONS</td><td>String</td><td colspan="3">NA</td><td></td></tr>
</table>

LOCAL-TIME	String	14	
LOCAL-DATE	String	14	
LIST-OF-SESSION-KEYS	String	NA	
TIME-SYNCHRONIZATION-RECIPIENTS	String	9 + (5 * Number of Time Synchronization Recipients count)	
DEVICE-ADDRESS-BINDING	String	51	
CONFIGURATION-FILES	String	NA	
LAST-RESTORE-TIME	String	21	
ACTIVE-COV-SUBSCRIPTIONS	String	NA	
SLAVE-PROXY-ENABLE	String	9 + (Number of Slave Proxy Enable count)	
MANUAL-SLAVE-ADDRESS-BINDING	String	9 + (9 * Number of Manual Slave Address Binding count)	
AUTO-SLAVE-DISCOVERY	String	9 + (Number of Auto Slave Discovery count)	
SLAVE-ADDRESS-BINDING	String	9 + (9 * Number of Slave Address Binding count)	
INACTIVE-TEXT	String	12 + (Size of Inactive Text in characters)	
ACTIVE-TEXT	String	12 + (Size of Active Text in characters)	
CHANGE-OF-STATE-TIME	String	19	
TIME-OF-STATE-COUNT-RESET	String	19	
TIME-OF-ACTIVE-TIME-RESET	String	19	
EFFECTIVE-PERIOD	String	19	
SCHEDULE-DEFAULT	String	NA	
LIST-OF-OBJECT-PROPERTY-REFERENCES	String	9 + (15 * Number of List Of Object Property References count)	
FILE-TYPE	String	12 + (Size of File Type in characters)	
MODIFICATION-DATE	String	NA	
DATE-LIST	String	9 + (12 * Number of Date List count)	
SCALE	String	14	
PRESCALE	String	15	
VALUE-CHANGE-TIME	String	NA	
LOGGING-RECORD	String	NA	
LOGGING-OBJECT	String	NA	
DESCRIPTION-OF-HALT	String	12 + (Size of Description Of Halt in characters)	
PROGRAM-LOCATION	String	12 + (Size of Program Location in characters)	
INSTANCE-OF	String	12 + (Size of Instance Of in characters)	

Property	**WEEKLY-SCHEDULE**										
Value Format	String										
Maximum Length (In Bytes)	11+		**Data Type**	**Time + Value**							
				Mon	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**	**Total**
		Per count in the day	**Null**	1	1	1	1	1	1	1	7
			Bool	6	6	6	6	6	6	6	42
			UInt	10	10	10	10	10	10	10	70
			Int	10	10	10	10	10	10	10	70
			Real	10	10	10	10	10	10	10	70
			Double	-	-	-	-	-	-	-	NA
			Octet String	8 + (Size Octet String in characters / 2)	8 + (Size Octet String in characters / 2)	8 + (Size Octet String in characters / 2)	8 + (Size Octet String in characters / 2)	8 + (Size Octet String in characters / 2)	8 + (Size Octet String in characters / 2)	8 + (Size Octet String in characters / 2)	…
			Char String	8 + (Size Char String in characters)	8 + (Size Char String in characters)	8 + (Size Char String in characters)	8 + (Size Char String in characters)	8 + (Size Char String in characters)	8 + (Size Char String in characters)	8 + (Size Char String in characters)	…
			Bit String	7 + (Size Bit String in characters / 8)	7 + (Size Bit String in characters / 8)	7 + (Size Bit String in characters / 8)	7 + (Size Bit String in characters / 8)	7 + (Size Bit String in characters / 8)	7 + (Size Bit String in characters / 8)	7 + (Size Bit String in characters / 8)	…
			Enum	7 + (Enum in hex / 2)	7 + (Enum in hex / 2)	7 + (Enum in hex / 2)	7 + (Enum in hex / 2)	7 + (Enum in hex / 2)	7 + (Enum in hex / 2)	7 + (Enum in hex / 2)	…
			Date	10	10	10	10	10	10	10	70
			Time	10	10	10	10	10	10	10	70
			Object ID	10	10	10	10	10	10	10	70
Comments											

Property	**EXCEPTION-SCHEDULE**				
Value Format	String				
Maximum Length (In Bytes)	18+	**Per count of Exception Schedule**	**Event Priority**	**Data Type**	**Per count of the List of Time Values (Period + Value)**
			4	**Null**	11
				Bool	11
				UInt	15
				Int	15
				Real	15
				Double	NA
				Octet String	13 + (Size Octet String in characters / 2)
				Char String	13 + (Size Char String in characters)
				Bit String	12 + (Size Bit String in characters / 8)

				Enum	12 + (Enum in hex / 2)
				Date	15
				Time	15
				Object ID	15
Comments					

Datatype Table:

Datatype Number	Datatype description	Example of usage with value
0	NULL	0:NULL
1	Boolean	1:1 (1 is true, 0 is false)
2	Unsigned	2:45
3	Signed	3:-34
4	Real	4:5.6
5	Double	5:5666
6	Octect String	6:12 23 34
7	String	7:Test
8	Bit String	8:10101010
9	Enumerated	9:1
10	Date	10:21-01-2013
11	Time	11:12:12:12.5
12	Object Identifier	12:1:1 1:1 (analog output,instance 1) See table ObjectIdentifier definitions

Object Identifier Definitions Reference:

ACCUMULATOR	23
ANALOG_INPUT	0
ANALOG_OUTPUT	1
ANALOG_VALUE	2
AVERAGING	18
BINARY_INPUT	3
BINARY_OUTPUT	4
BINARY_VALUE	5
CALENDAR	6
COMMAND	7
DEVICE	8
EVENT_ENROLLMENT	9
FILE	10
GROUP	11
LIFE_SAFETY_POINT	21
LIFE_SAFETY_ZONE	22
LOOP	12
MULTI_STATE_INPUT	13
MULTI_STATE_OUTPUT	14
MULTI_STATE_VALUE	19
NOTIFICATION_CLASS	15
PROGRAM	16
PULSE_CONVERTER	24

```
SCHEDULE  17
TREND_LOG  20
```

BACnetPriorityArray:

The BacnetPriorityArray data type is represented as an array of 16 priority values. The Bacne support Null values and Unsigned values on these fields.
To write on a priority array data type you must write a Present-Value specifying the priority, please look at the page 28 for more information.
The read operations can be performed directly on the property of this type. The format for displaying this datatype is:

`<Value for Priority1>|< Value for Priority2>|< Value for Priority3>|. . .|< Value for Priority16>`

`Example:   NULL|NULL|NULL|NULL|NULL|NULL|12|NULL|`

The meaning of each priority level is described on the table below.

Priority Level	Application	Priority Level	Application
1	Manual-Life Safety	9	Available
2	Automatic-Life Safety	10	Available
3	Available	11	Available
4	Available	12	Available
5	Critical Equipment Control	13	Available
6	Minimum On/Off	14	Available
7	Available	15	Available
8	Manual Operator	16	Available

Source: *Bacnet Standard Specification*

BACnetDateTime

The BacnetDateTime is represented as an date and a time, following this format:

dd-mm-yyyy hh:mm:ss.ms

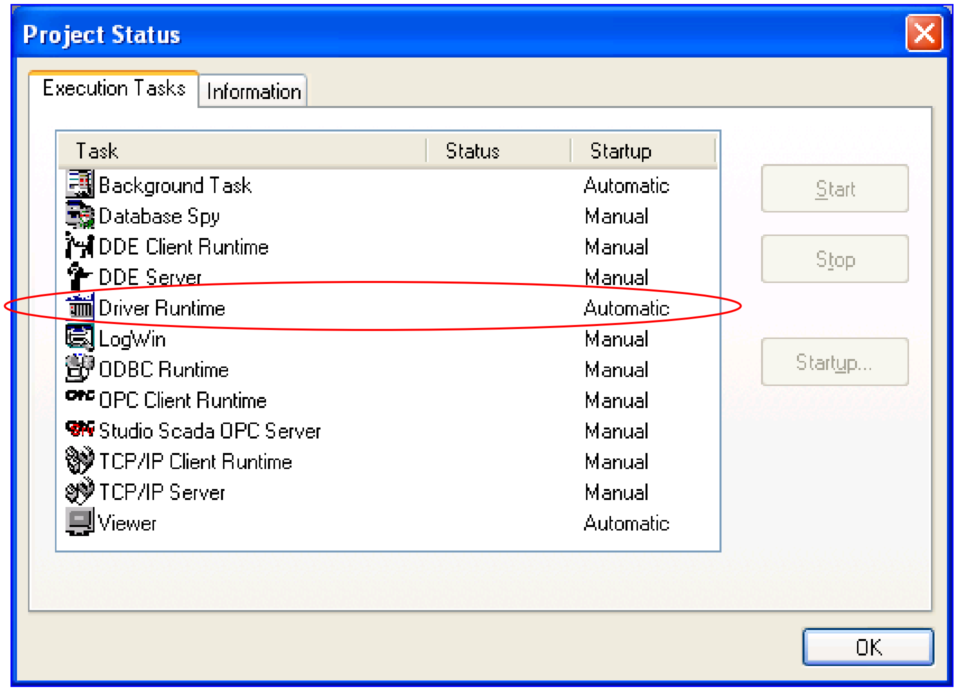
Example: 12-12-2012 12:12:12.4

Executing the Driver

By default, Studio will automatically execute your selected communication driver(s) during application runtime. However, you may verify your application's runtime execution settings by checking the *Project Status* dialog.

To verify that the communication driver(s) will execute correctly:

1. From the main menu bar, select **Project** → **Status**. The *Project Status* dialog displays:



Task	Status	Startup
Background Task		Automatic
Database Spy		Manual
DDE Client Runtime		Manual
DDE Server		Manual
Driver Runtime		Automatic
LogWin		Manual
ODBC Runtime		Manual
OPC Client Runtime		Manual
Studio Scada OPC Server		Manual
TCP/IP Client Runtime		Manual
TCP/IP Server		Manual
Viewer		Automatic

Project Status Dialog

2. Verify that the *Driver Runtime* task is set to **Automatic**.

    - If the setting is correct, then proceed to step 3 below.
    - If the **Driver Runtime** task is set to **Manual**, then select the task and click the **Startup** button to toggle the task's *Startup* mode to **Automatic**.

3. Click **OK** to close the *Project Status* dialog.

4. Start the application to run the driver.

Troubleshooting

If the BACNE driver fails to communicate with the target device, then the database tag(s) that you configured for the **Read Status** or **Write Status** fields of the Main Driver Sheet will receive an error code. Use this error code and the following table to identify what kind of failure occurred.

Error	Description	Possible Causes	Procedure to Solve
0	OK	▪ Communication without problems	▪ None
1	Invalid operation	▪ Invalid Header or Address	▪ Check the header and address
2	Malformed packet	▪ Disconnected cables ▪ PLC turned off, in Stop or Error mode ▪ Wrong Station number ▪ Wrong RTS/CTS control settings	▪ Check cable wiring. ▪ Check PLC state. It must be RUN. ▪ Check station number. ▪ See "Link Characteristics" section for valid RTS/CTS configurations. Check required configuration.
3	Wrong link	▪ Wrong RTS/CTS control settings ▪ Wrong link setting on the driver settings station	▪ Check the driver settings station
4	Could not create receiver sink	▪ Restart the driver	▪ Restart the driver
5	Termination error	▪ Wrong Station number ▪ Invalid value in Station field	▪ Check the station number ▪ Check the Station field configuration
6	Unsupported BVLL function	▪ Disconnected cables ▪ PLC turned off, or in Stop or error mode ▪ Wrong Station number ▪ Wrong RTS/CTS control settings	▪ Check the cable wiring ▪ Check the PLC state (it must be RUN) ▪ Check the station number. ▪ Check the right configuration. Review the Communication Parameters section for valid RTS/CTS configurations.
7	Segmentation is not supported	▪ Too many properties on the Standard Driver Sheet	▪ Split it in many Standard Driver Sheet or use the Main Driver Sheet
8	Unknown APDU type	▪ Invalid Request	▪ X
9	Object not found	▪ The requested object is not found on the device.	▪ Check on the device if the object exists
10	Invalid station on driver settings	▪ The format of the station is incorrect	▪ See the section of station to learn how to validate the station.
11	Invalid MAC on driver settings	▪ The MAC specified on the driver settings is invalid	▪
12	Invalid UDP Port on driver settings	▪ Port number is invalid	▪ Check the port number of the driver
13	Device is unavailable	▪ Device is not accessible	▪ Check the device status
14	Unknown object	▪ The object does not exists	▪ Check the objects on the device ▪ Check the information on the Output Window (Enable the protocol analyzer) it will show exactly what is wrong. Example:
15	Unknown property	▪ The property does not exists	▪ Check the properties on the device

Error	Description	Possible Causes	Procedure to Solve
16	Write denied	▪ You are not allowed to write on the property	▪ Check the property is it is really writeable
17	Write error	▪ Error while writing	▪ Check the device status.
18	Invalid Format	▪ The value typed is invalid	▪ Check the format of the value typed
19	Unexpected error	▪ Device is not communicating as expected	▪ Check driver settings, check if device is online, restart the driver. Contact technical support if required.
20	Internal error	▪ Device is not communicating as expected	▪ Check driver settings, check if device is online, restart the driver. Contact technical support if required.

⇨ **Tip:**

You can monitor communication status by establishing an event log in Studio's *Output* window (*LogWin* module). To establish a log for **Field Read Commands**, **Field Write Commands** and **Protocol Analyzer,** right-click in the *Output* window and select the desired options from the pop-up menu.

You can also use the Remote *LogWin* module (**Tools** → **Remote LogWin**) to establish an event log on a remote unit.

If you are unable to establish communication between Studio and the target device, then try instead to establish communication using the device's own programming software. Quite often, communication is interrupted by a hardware or cable problem or by a device configuration error. If you can successfully communicate using the programming software, then recheck the driver's communication settings in Studio.

To test communication between Studio and the device, we recommend using the sample application provided rather than your new application.

If you must contact us for technical support, please have the following information available:

- **Operating System** (type and version): To find this information, select **Tools** → **System Information**.
- **Project Information**: To find this information, select **Project** → **Status.**
- **Driver Version** and **Communication Log**: Displays in the Studio *Output* window when the driver is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Sample Application

There was not an official sample application available for this driver by the time that this document was written.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of changes
A	1.00	Leandro Coeli	Dec 07, 2006	New version.
B	1.00	Michael D. Hayden	Jan 16, 2007	Edited for format, language, and usability.
C	1.00	Eric Vigiani	Mar 14, 2008	Doc. Revision
D	2.00	Eric Vigiani	Jun 20, 2008	Modified to work properly with NAE devices.
E	2.1	Paulo Balbino André Körbes	Jul 29, 2010	Added Schedule Object Added Data Types section
F	3.0	Paulo Balbino	Aug 26, 2011	Complete driver redesign
G	3.0	Paulo Balbino	Nov,1,2011	Added Priority for writing.
H	3.0	Paulo Balbino	Oct, 22,2012	Added more information about BACnetPriorityArray and BACnetDateTime
I	3.1	Paulo Balbino	Jan, 21, 2013	Added more information about Exception Scheduler
J	3.2	Paulo Balbino	Aug, 23, 2013	Changed write priority Resolved Exception Scheduler issues Changed MSV and MSO present-value data type Resolved issue with the Station field not working with IP Address Updated documentation information for the Driver Settings window
K	3.3	Vijay Kankanala	Nov, 05, 2013	Added Life Safety Point Object Type (LSP). Added Life Safety Zone Object Type (LSZ).
L	3.4	Vijay Kankanala	Mar, 25, 2014	Exception Scheduler: Implemented Start & End Date format. Added Accumulator Object Type (ACC). Added Averaging Object Type (AVG). Added Calendar Object Type (CAL). Added Command Object Type (CMD).
M	3.5	Vijay Kankanala	May, 13, 2014	Added Event Enrollment Object Type (EVT). Added Loop Object Type (LOP). Added Notification Class Object Type (NOT).
N	3.6	Vijay Kankanala Felipe Andrade	Apr, 07, 2015	Added Program Object Type (PRG). Added Pulse Converter Object Type (PC). Added Group Object Type (GRP). Added File Object Type (FO). Added Trend Log Object Type (TL). Changed datatype for SCHEDULE-DEFAULT to ANY Supported LIST-OF-OBJECT-PROPERTY-REFERENCES for Schedule Object Type (SCH). Fixed documentation. Updated the Property Block Size Table.
O	3.7	Paulo Balbino	April, 22,2015	Updated interfaces information
P	3.8	Anushree Phanse	Oct 27, 2015	Fixed OBJECT-NAME, INACTIVE TEXT, ACTIVE TEST, DESCRIPTION etc. to support umlauts, such as ü, ä or ö (ANSI,

				UTF-8 etc.).
Q	3.9	Anushree Phanse	Apr, 28, 2016	Changed datatype for BO BI BV present value property from real to Integer Fixed issue with Present-Value with Priority Array Index group split causing tag quality to go bad.
R	3.9	Anushree Phanse	July 20, 2016	Removed references to MS/TP support, no change in the driver.
S	3.9	Vamsikrishna Atluri	Aug 18, 2016	Corrected the documentation on the Destination MAC Address length, no change in the driver Added a syntax <BacNet IP Address>:<SubNet>:<Device ID>:<MAC Address> in the Driver Station Configuration, no change in the driver
T	3.10	Anushree Phanse	Aug 23, 2016	Fixed the issue of priority arrays showing ‘0’ instead of ‘NULL’ Fixed the issue of tag values crossing
U	3.10	Anushree Phanse	Jan 04, 2017	Improved driver documentation for using station with the parameter of BACnet MAC Address. No change in the driver.
V	3.11	Anushree Phanse	Feb 08 2017	Fixed the issue where the OI.BACLITE (BACNE driver) would crash when communicating in a specific condition.