BACnet Master

User Manual

A-BNM-B

A-BNM-P

A-BNM-M

Document No. D172-010
11/2026
Revision 1.1

aparian

CONTENTS

1. Preface 8
1.1. Introduction to the BACnet Master 8
1.2. Features 10
1.3. Architecture 11
1.4. Additional Information 14
1.5. Support 14
2. Installation 15
2.1. Module Layout 15
2.2. Module Mounting 16
2.3. Power 18
2.4. RS232 / RS485 Ports 18
2.5. RS485 Termination 19
2.6. Ethernet Ports 19
2.7. North American Hazardous Location Approval 20
3. Setup 21
3.1. Install Configuration Software 21
3.2. Network Parameters 21
3.3. Creating a New Project 26
3.4. General parameters 28
3.5. BACnet Configuration 31
3.6. Serial Configuration 33
3.7. Primary Interface 34
3.7.1. EtherNet/IP Target 34
3.7.1.1. Studio / Logix 5000 Configuration 34
3.7.1.2. Internal Data Space Mapping 43
3.7.2. Modbus (Server and Client) 44
3.7.2.1. Internal Data Space Mapping 46
3.7.2.2. Modbus Auxiliary Map 48
3.7.3. EtherNet/IP Originator 50
3.7.3.1. EtherNet/IP Class 1 Device Connections 50
3.7.3.2. EtherNet/IP Explicit Message Device Connections 59

3.7.3.3. Internal Data Space Mapping 63
3.8. BACnet Devices 65
3.8.1. Manual Addition 65
3.8.1.1. BACnet Device Configuration – General 66
3.8.1.2. BACnet Device – Property Configuration 68
3.8.2. Device Discovery 70
3.8.2.1. Discovery Instantiation Options 72
3.8.2.2. Scan 73
3.8.2.3. Discovery Device Functions 76
3.9. Internal Data Space Map 82
3.9.1. Copy From 83
3.9.1.1. Internal 83
3.9.1.2. EIP Target 83
3.9.1.3. EIP Originator 84
3.9.1.4. Modbus Register 85
3.9.1.5. BACnet Device 86
3.9.1.6. System 86
3.9.2. Copy To 87
3.9.2.1. Internal 87
3.9.2.2. EIP Target 87
3.9.2.3. EIP Originator 88
3.9.2.4. Modbus Register 88
3.9.2.5. BACnet Device 89
3.9.2.6. System 90
3.10. Constants 90
3.11. Advanced 91
3.12. Module Download 93
4. SD Card 95
4.1. Firmware 95
4.2. Network Parameters 96
4.3. Configuration 97
4.3.1. Manual Copy 98
4.3.2. Slate Triggered Upload 99

5. Device Firmware Update 101
6. Security 104
6.1. Securing Process 105
6.1.1. Configuration 105
6.1.1.1. General Configuration 106
6.1.1.2. Password Rule Configuration 108
6.1.1.3. Custom Users Configuration 109
6.1.1.4. Services Configuration 110
6.1.1.5. IP Access Control List Configuration 112
6.1.1.6. MAC Access Control List Configuration 113
6.1.2. Download Security Configuration 115
6.1.3. Initialize User Authentication 116
6.2. Operation 119
6.2.1. Initial User Login 119
6.2.2. Role Specific Operations 121
6.2.2.1. Security Status 121
6.2.3. Change Other Password 126
6.2.4. Clear All Security 127
6.3. Event Log 128
6.4. Reset Security Configuration 128
7. Operation 129
7.1. BACnet 129
7.1.1. Input Data 129
7.1.1.1. Read 129
7.1.1.2. COV (Change of State Subscription) 130
7.1.1.3. COV Prop (Change of State Property Subscription) 130
7.1.1.4. COV Prop Inc (Change of State Property Subscription with Increment) 131
7.1.1.5. Map 131
7.1.2. Output Data 132
7.1.2.1. Write 132
7.1.2.2. Write Change 133
7.1.3. Format 133
7.1.3.1. Standard 133

7.1.3.2. w/Status (Standard With Status) 133
7.1.3.3. Packed 134
7.2. EtherNet/IP Target 134
7.2.1. Class 1 Assembly Mapping 135
7.3. EtherNet/IP Originator 137
7.3.1. EtherNet/IP Class 1 Connections 137
7.3.2. Explicit EtherNet/IP Messaging 138
7.4. Modbus 139
7.5. Data Formats 140
7.5.1. System Status 140
7.5.2. System Control 141
7.5.3. Device Status 141
7.5.4. Property Status 142
8. Diagnostics 143
8.1. LEDs 143
8.2. Module Status Monitoring in Slate 144
8.2.1. General 145
8.2.2. BACnet MS/TP Statistics 148
8.2.3. BACnet /Ethernet Statistics 149
8.2.4. BACnet /IP Statistics 150
8.2.5. EtherNet/IP Explicit 151
8.2.6. EtherNet/IP Map 152
8.2.7. EtherNet/IP Originator 153
8.2.8. Logix 154
8.2.9. Modbus 156
8.2.10. Modbus Auxiliary Items 157
8.2.11. CIP Statistics 158
8.2.12. Ethernet Clients 158
8.2.13. TCP/ARP 159
8.3. EtherNet/IP Target Device Status Monitoring In Slate 159
8.3.1.1. General 160
8.3.1.2. Input Data 161
8.3.1.3. Output Data 162

8.4. BACnet Device Monitoring 162
8.4.1.1. General 163
8.4.1.1. Properties 164
8.5. Module Event Log 165
8.6. BACnet Packet Capture 166
8.7. BACnet Explicit Message Utility 168
8.8. Modbus Packet Capture 169
8.9. Module Status Report 171
8.10. Modbus Summary CSV 172
8.11. Modbus Expanded CSV 173
8.12. Internal Register Viewer 174
9. BACnet Object Properties 176
9.1. Supported BACnet Objects 176
9.2. Supported BACnet Object Properties 176
10. Technical Specifications 187
10.1. Dimensions 187
10.1. Electrical 188
10.2. Environmental 188
10.3. Ethernet 189
10.4. Serial Port (RS232) 189
10.5. Serial Port (RS485) 190
10.6. BACnet 190
10.7. EtherNet/IP Target 191
10.8. EtherNet/IP Originator 191
10.9. Modbus Client 191
10.10. Modbus Server 192
10.11. Certifications 192
11. Index 194

Revision History

Revision	Date	Comment
1.0	7 September 2026	Initial document
1.1	17 November 2026	Update Ethernet Specification

1. PREFACE

1.1. INTRODUCTION TO THE BACNET MASTER

This manual describes the installation, operation, and diagnostics of the Aparian BACnet Master (A-BNM). The BACnet Master provides flexible integration between BACnet devices and EtherNet/IP and Modbus (Modbus-RTU232, Modbus-RTU485, and Ethernet Modbus-TCP) devices.

Communication with BACnet devices can be via Ethernet (BACnet/IP and BACnet/Ethernet) and RS485 (BACnet MS/TP).

The BACnet Master allows the user to integrate Modbus devices into a Rockwell Automation Logix platform (ControlLogix or CompactLogix).

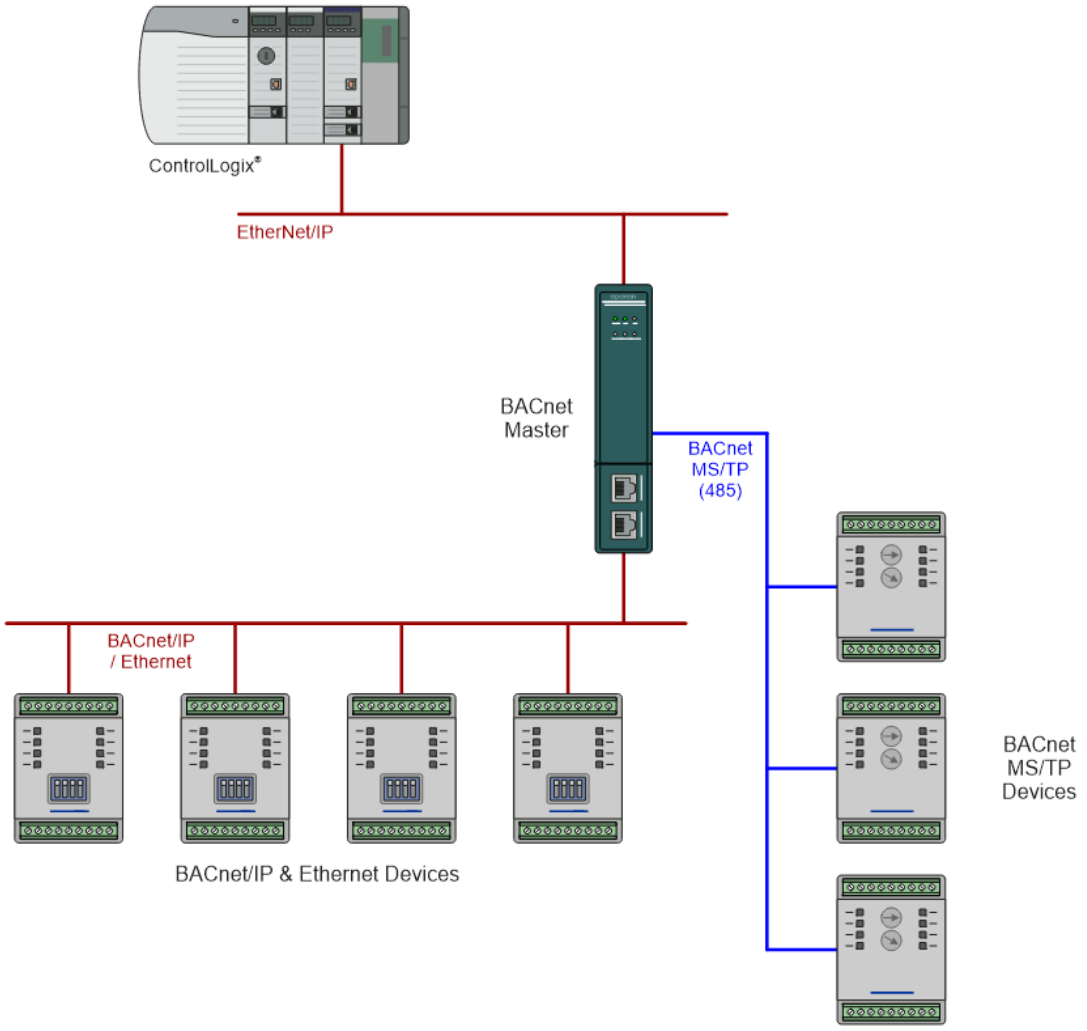


Figure 1.1. – BACnet System

Primary Interface:

The module can operate in one of four Primary Interface modes:

EtherNet/IP Target
Here a remote EtherNet/IP device (e.g. a Logix controller) establishes a number of Class 1 connections to the module. BACnet data can be mapped into a maximum of 25 separate input and output class 1 cyclic connections to the Logix controller (allowing up to 12KB input and 12KB output to be exchanged at the requested packet interval – RPI).

EtherNet/IP Originator
As an EtherNet/IP originator, the BACnet master can use one of three methods to exchange data:

- **EtherNet/IP Explicit Messaging**

  Here the BACnet master exchanges data with up to 10 EtherNet/IP devices using explicit messaging. The module can use either Class 3 or Unconnected Messaging (UCMM) to Get and Set data in the remote EtherNet/IP devices.

- **Direct-To-Tag technology**

  Here the BACnet master exchanges data with a Logix controller without the need to write any application code (e.g. ladder) in Studio 5000. The BACnet data is directly read from, or written to, Logix tags.

- **EtherNet/IP Class 1 connection**

  BACnet data can be mapped to a maximum of 10 EtherNet/IP devices using input and output class 1 cyclic connections. This will allow the BACnet Master to "own" the EtherNet/IP target device and exchange BACnet device data using the EtherNet/IP device's input and output assemblies.

Modbus Server
As a Modbus Server, the BACnet data will be written to, or read from, the module's internal Modbus Registers (Holding or Input Registers). These registers can be accessed by a remote Modbus Client using Modbus TCP, RTU232, and RT485.

Modbus Client
As a Modbus Client, the BACnet data will be written to, or read from, the module's internal Modbus Registers (Holding or Input Registers). The Modbus Auxiliary Map can then be used to configure the Modbus data exchange between multiple remote Modbus Server devices and the module's internal Modbus registers via Modbus TCP, RTU232, and RT485.

The BACnet Master is configured using the Aparian Slate application. This program can be downloaded from www.aparian.com free of charge.

The module also provides a range of statistics, BACnet and Modbus packet capture functions, and internal Modbus and Data table reads to simplify the diagnostic process for remote diagnosis.

The module has two Ethernet ports and supports Device-Level-Ring (DLR) architectures.

1.2. FEATURES

- Support up to:
  - 200 BACnet data points with 50 BACnet devices (A-BNM-B)
  - 1000 BACnet data points with 250 BACnet devices (A-BNM-P)
  - 2500 BACnet data points with 250 BACnet devices (A-BNM-M)
- Support up to 256 Properties per BACnet device.
- Supports BACnet communication on Ethernet and RS485.
- Logix L5X generation - Ready for import
- Supports Passthrough Messaging from Logix
- Module has various Primary Interfaces:
  - EtherNet/IP Target (Class 1 connection)
  - Modbus Server (TCP, RTU232, and RTU485)
  - Modbus Client (TCP, RTU232, and RTU485)
  - EtherNet/IP Originator (Class 1 connection with up to 10 EtherNet/IP devices and Explicit Messaging, including Direct-To-Tag Logix tag access, with up to 10 EtherNet/IP devices).
- Slate software provides a BACnet and Modbus packet capture utility for better diagnosis of communication issues.
- Dual Ethernet ports which support Device-Level-Ring (DLR).
- Network Time Protocol (NTP) supported for external time synchronization.
- Small form factor – DIN rail mounted.

The security services provide the following features:

- Configurable level of security
- Encrypted EtherNet/IP communication
- Fixed User Roles and Custom Users
- Configurable Password rules
- Login Expiry
- Failed Login Cool-off and Account Disable options
- Configurable Global Services and Custom Ports
- Optional IP Address Access Control List
- Optional MAC Access Control List
- Restrictions on Reset commands

- Restrictions on Device Flash options
- Security Audit trail in Event Log

1.3. ARCHITECTURE

The figures below provide an example of the typical network setup for connecting BACnet devices to either EtherNet/IP or Modbus TCP.

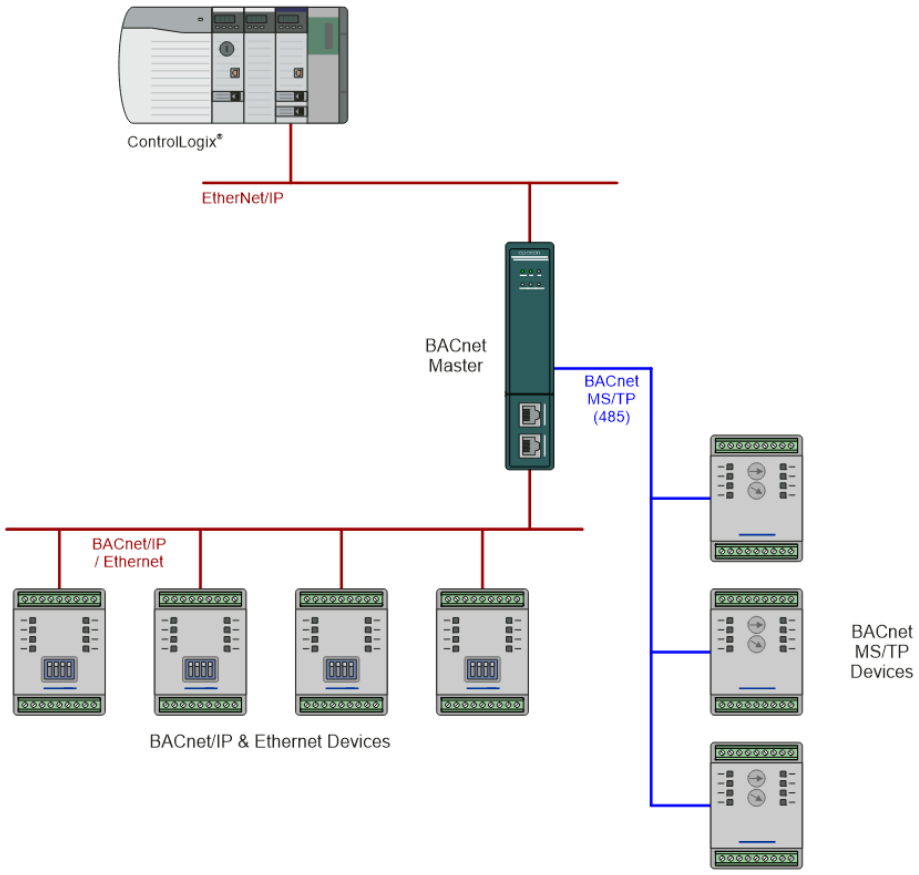


Figure 1.2. – Example of connecting to a Logix Controller

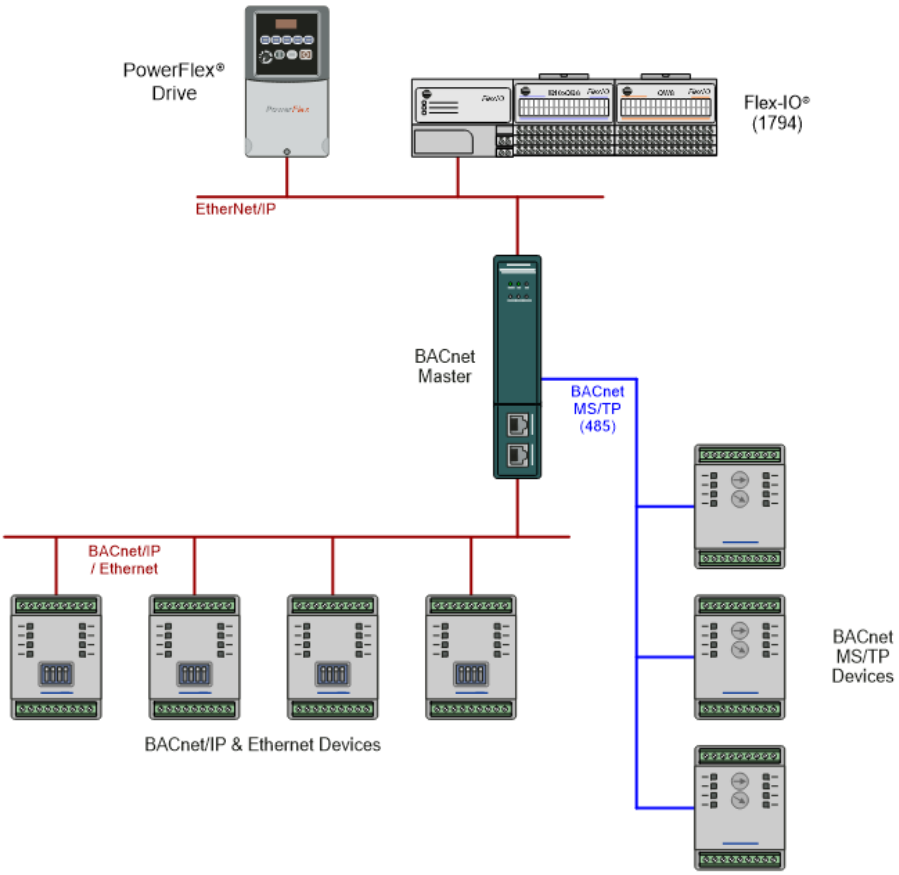


Figure 1.3. – Example of connecting EtherNet/IP I/O

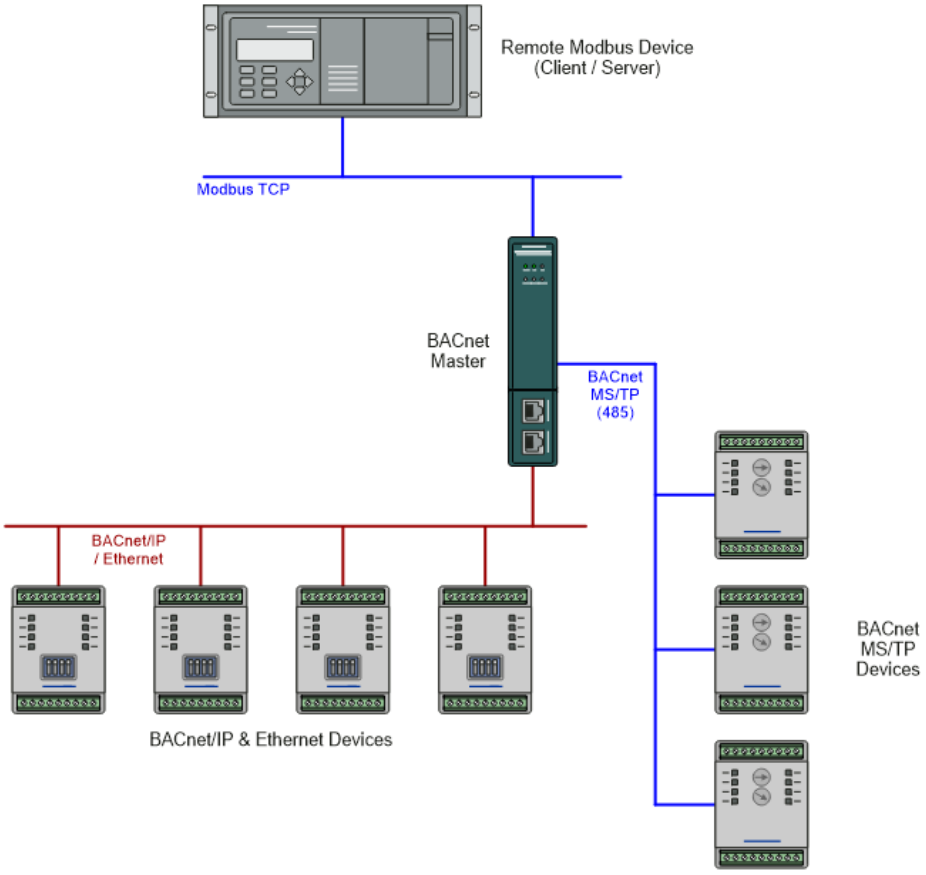


Figure 1.4. - Example of connecting to a Modbus TCP Client or Server

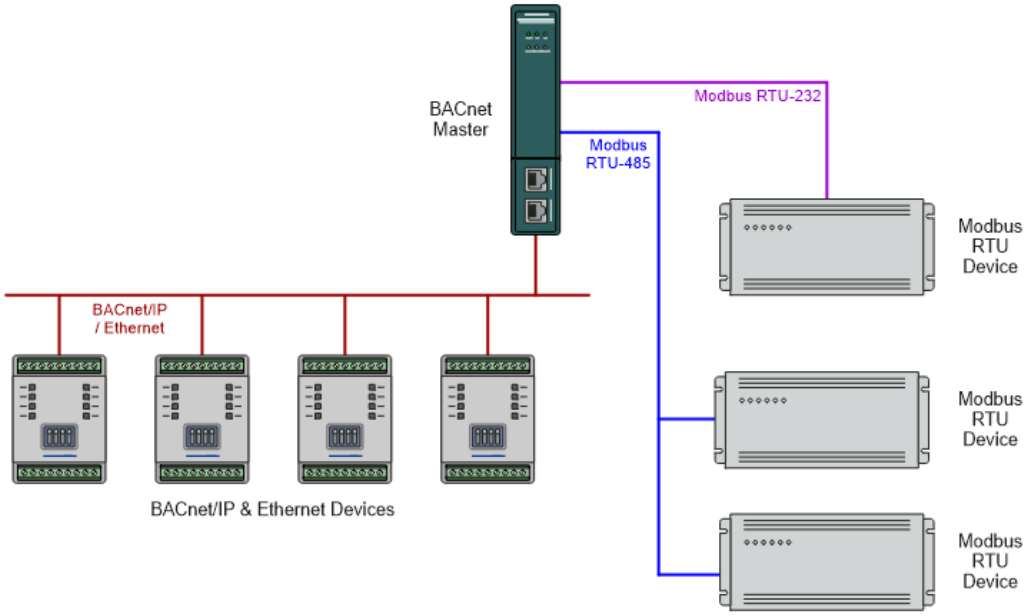


Figure 1.5. - Example of connecting to a Modbus RTU Client or Server

1.4. ADDITIONAL INFORMATION

The following documents contain additional information that can assist the user with the module installation and operation.

Resource	Link
Slate Installation	http://www.aparian.com/software/slate
BACnet Master User Manual BACnet Master Datasheet Example Code & UDTs	https://www.aparian.com/products/BACnetMaster
Ethernet wiring standard	www.cisco.com/c/en/us/td/docs/video/cds/cde/cde205_220_420/installation/guide/cde205_220_420_hig/Connectors.html

Table 1.1. - Additional Information

1.5. SUPPORT

Technical support is provided via the Web (in the form of user manuals, FAQ, datasheets etc.) to assist with installation, operation, and diagnostics.

For additional support the user can use either of the following:

Resource	Link
Contact Us web link	https://aparian.com/aparian-tech-support-request
Support email	support@aparian.com

Table 1.2. – Support Details

2. INSTALLATION

2.1. MODULE LAYOUT

The module has two ports at the bottom of the enclosure as shown in the figure below. The ports are used for RS232 and RS485 serial, and power.

The power port uses a three-way connector which is used for the DC power supply positive and negative (or ground) voltage as well as the earth connection.

The Ethernet cable used for the Ethernet ports must be wired according to industry standards which can be found in the additional information section of this document.

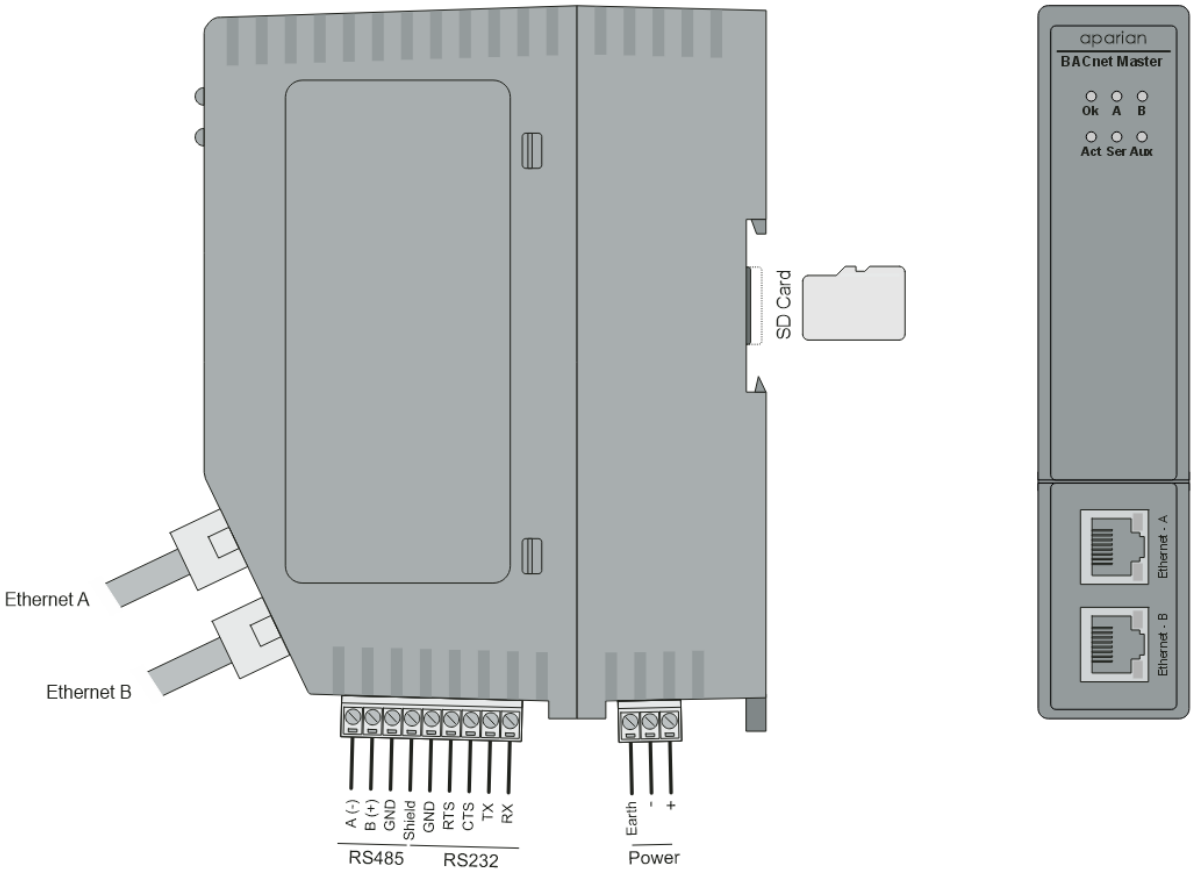


Figure 2.1 – BACnet Master side and front view

The module also supports an SD Card for disaster recovery which can be used to automatically update the configuration, network parameters, and/or firmware of a new module.

The module provides six diagnostic LEDs as shown in the front view figure above. These LEDs are used to provide information regarding the module system operation, the Ethernet interface, serial interfaces and auxiliary communication.

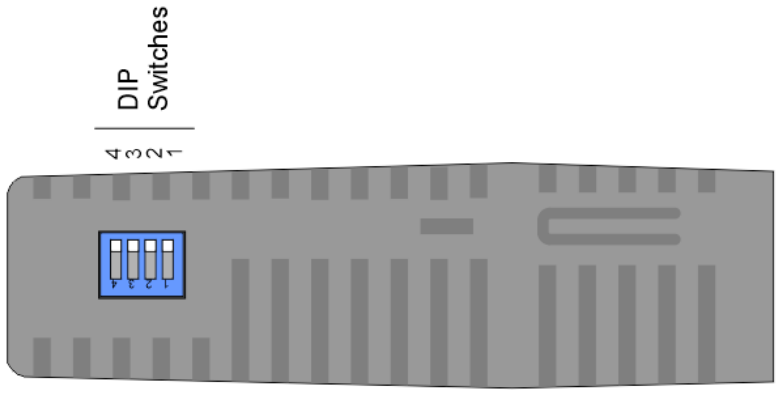


Figure 2.2 – BACnet Master top view

The module provides four DIP switches at the top of the enclosure as shown in the top view figure above.

DIP Switch	Description
DIP Switch 1	Used to force the module into “Safe Mode”. When in “Safe Mode” the module will not load the application firmware and will wait for new firmware to be downloaded. This should only be used in the rare occasion when an earlier firmware update was interrupted at a critical stage.
DIP Switch 2	This will force the module into DHCP mode which is useful when the user has forgotten the IP address of the module.
DIP Switch 3	This DIP Switch is used to lock the configuration from being overwritten by the Slate. When set Slate will not be able to download to the module.
DIP Switch 4	When this DIP Switch is set at bootup it will force the module Ethernet IP address to **192.168.1.100** and network mask 255.255.255.0. The user can then switch the DIP switch off and assign the module a static IP address if needed.

Table 2.1 - DIP Switch Settings

2.2. MODULE MOUNTING

NOTE: This module is an open-type device and is meant to be installed in an enclosure suitable for the environment such that the equipment is only accessible with the use of a tool.

The module provides a DIN rail clip to mount onto a 35mm DIN rail.

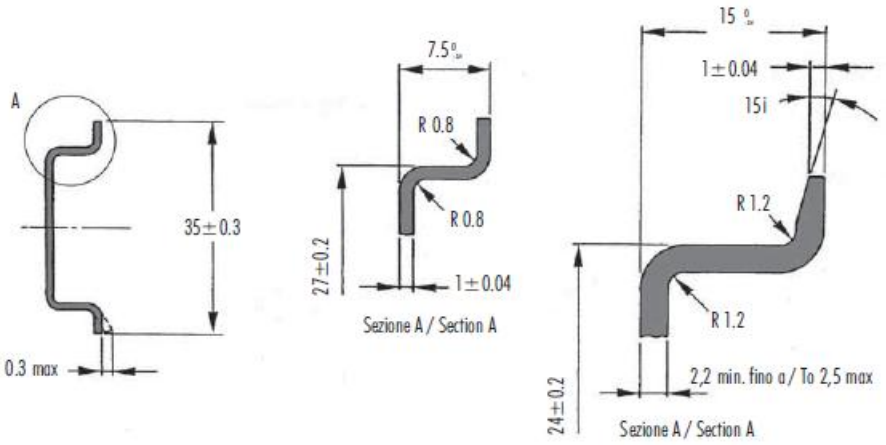


Figure 2.3 - DIN rail specification

The DIN rail clip is mounted on the bottom of the module at the back as shown in the figure below. Use a flat screwdriver to pull the clip downward. This will enable the user to mount the module onto the DIN rail. Once the module is mounted onto the DIN rail the clip must be pushed upwards to lock the module onto the DIN rail.

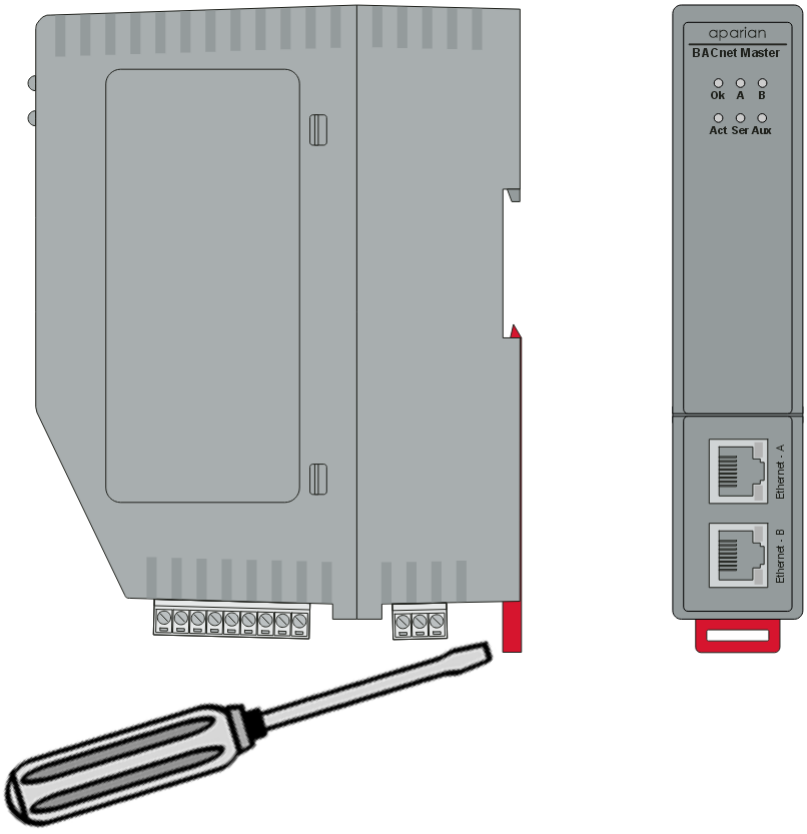


Figure 2.4 - DIN rail mounting

2.3. POWER

A three-way power connector is used to connect Power+, Power– (GND), and earth. The module requires an input voltage of 10 – 32 Vdc. **Refer** to the technical specifications section in this document.

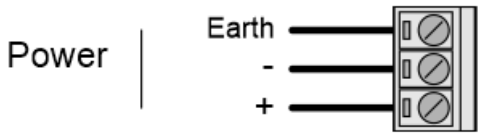


Figure 2.5 - Power connector

2.4. RS232 / RS485 PORTS

The nine-way connector is used to connect the RS232 and RS485 conductors for serial communication. The shield terminal can be used for shielded cable in high noise environments.

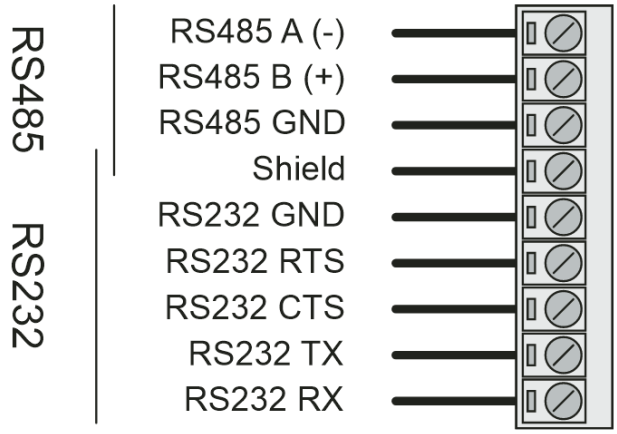


Figure 2.6 - RS232/RS485 connector.

The RS485 port provides the standard ***A(-)***, ***B(+)*** conductors.

For long RS485 networks, it is recommended to also connect the RS485 Ground ***(GND)*** reference.

The RS232 port provides the standard communication pins (***RX***, ***TX***, and ***GND***) as well as hardware handshaking lines for legacy systems (***RTS*** – Request to Send, ***CTS*** – Clear to Send).

Both RS232 and RS485 share a common cable shield connection which should be connected to the shield of the outgoing cable (RS232 and/or RS485).



NOTE: The shield of the RS232/RS485 port is internally connected to the power connector earth. When using a shield, it is important to connect the Earth terminal on the power connector to a clean earth. Failing to do this can lower the signal quality of the RS232/RS485 communication.



NOTE: When using a shielded cable, it is important that only one end of the shield is connected to earth to avoid current loops. It is recommended to connect the shield to the BACnet Master module rather than earthing it at the remote devices.

2.5. RS485 TERMINATION

All RS485 networks need to be terminated at the extremities (start- and end-points) of the communication cable. The termination for the RS485 network can be enabled/disabled via the module configuration. Enabling the termination will connect an internal 125 Ohm resistor across the positive (+) and negative (-) conductors of the RS485 network.

2.6. ETHERNET PORTS

The Ethernet connectors should be wired according to industry standards. **Refer** to the additional information section in this document for further details. The module has an embedded switch connecting the two Ethernet ports.

2.7. NORTH AMERICAN HAZARDOUS LOCATION APPROVAL

The following information applies when operating the product in hazardous locations:

SUITABLE FOR USE IN CLASS I, DIVISION 2, GROUPS A, B, C AND D HAZARDOUS LOCATIONS, OR NONHAZARDOUS LOCATIONS ONLY.

-20°C to +70°C, T5
Class 1, Div 2 Gps A, B, C, D

THIS EQUIPMENT IS OPEN-TYPE DEVICE AND IS MEANT TO BE INSTALLED IN AN ENCLOSURE SUITABLE FOR THE ENVIRONMENT SUCH THAT THE EQUIPMENT IS ONLY ACCESSIBLE WITH THE USE OF A TOOL.

WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT WHILE THE CIRCUIT IS LIVE OR UNLESS THE AREA IS KNOW TO BE FREE OF IGNITABLE CONCENTRATIONS.

WARNING - EXPLOSION HAZARD - SUBSTITUTION OF ANY COMPONENT MAY IMPAIR SUITABILITY FOR CLASS I, DIVISION 2.

3. SETUP

3.1. INSTALL CONFIGURATION SOFTWARE

All the network setup and configuration of the module is achieved by means of the Aparian Slate device configuration environment. This software can be downloaded from http://www.aparian.com/software/slate.

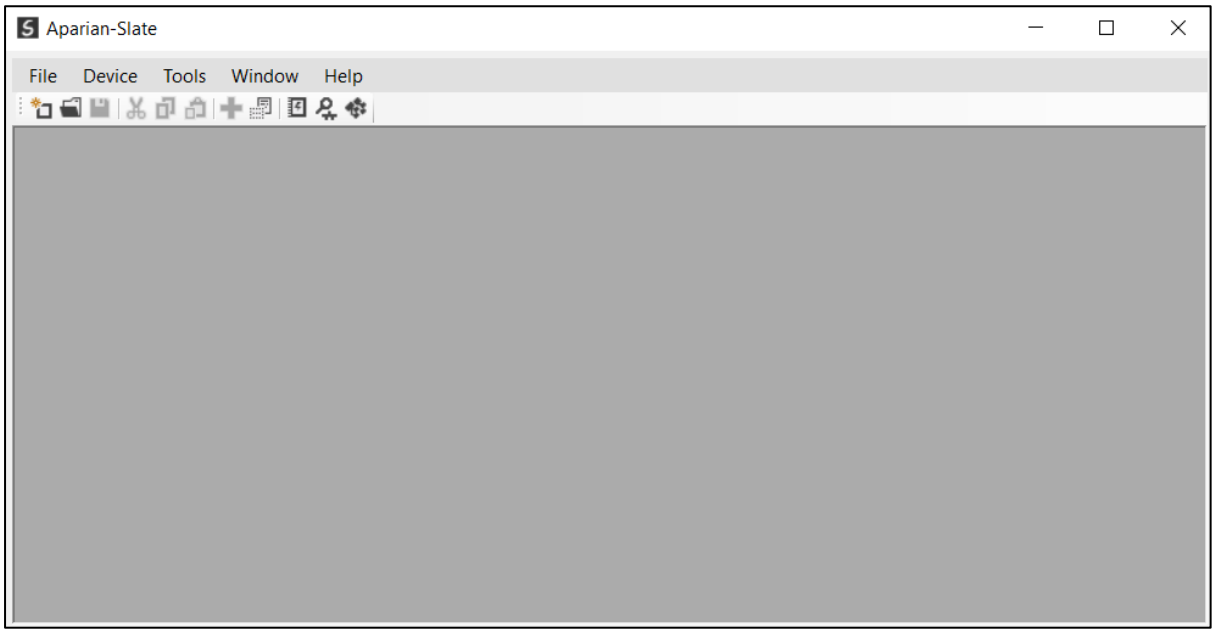


Figure 3.1. - Aparian Slate Environment

3.2. NETWORK PARAMETERS

The module will have DHCP (Dynamic Host Configuration Protocol) enabled as factory default. Thus, a DHCP server must be used to provide the module with the required network parameters (IP address, subnet mask, etc.). There are a number of DHCP utilities available, however it is recommended that the DHCP server in Slate be used.

Within the Slate environment, the ***DHCP Server*** can be found under the ***Tools*** menu.

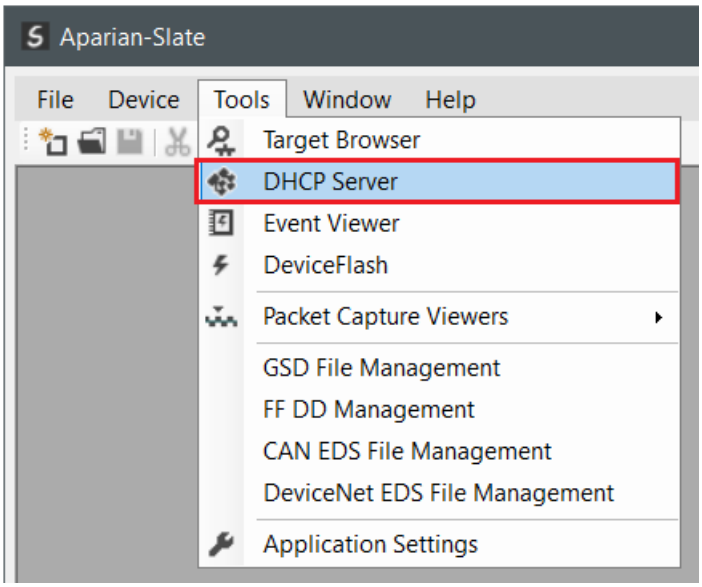


Figure 3.2. - Selecting DHCP Server

Once opened, the ***DHCP Server*** will listen on all available network adapters for DHCP requests and display their corresponding MAC addresses.

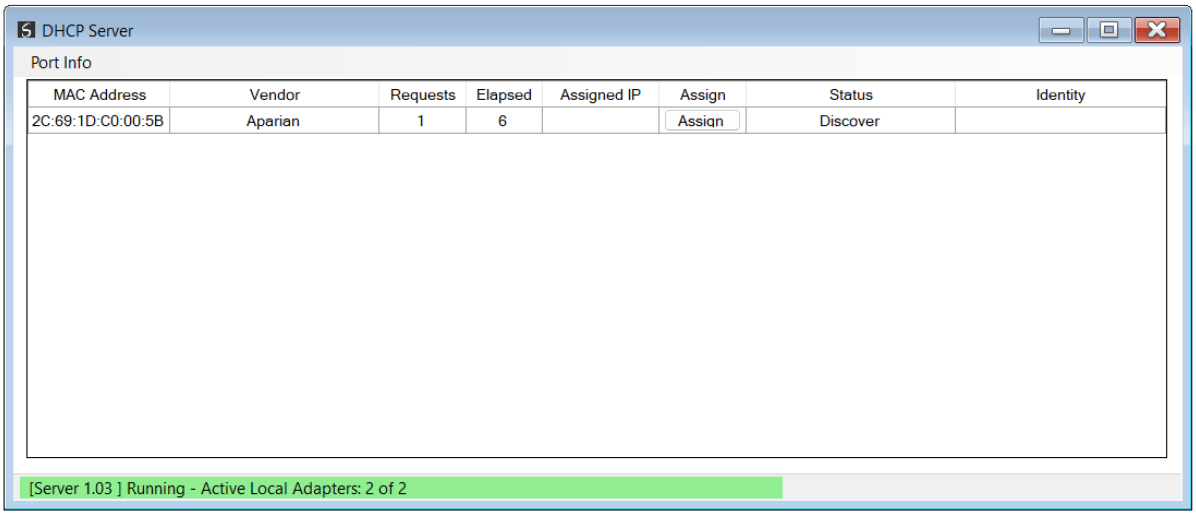


Figure 3.3. - DHCP Server

NOTE: If the DHCP requests are not displayed in the DHCP Server it may be due to the local PC's firewall. During installation, the necessary firewall rules are automatically created for the Windows firewall. Another possibility is that another DHCP Server is operational on the network, and it has assigned the IP address.

To assign an IP address, click on the corresponding "Assign" button. The IP Address Assignment window will open.

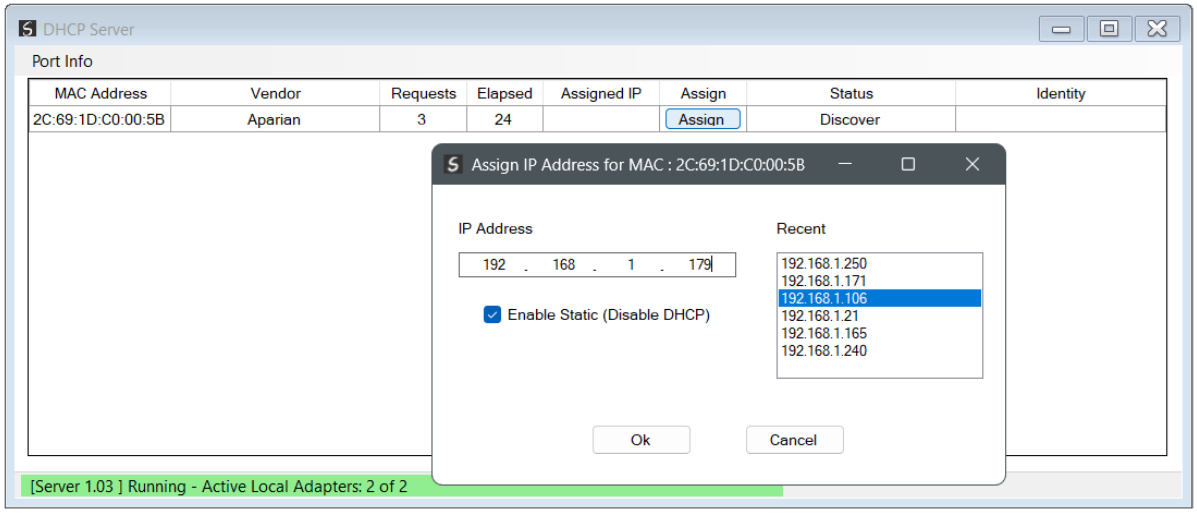


Figure 3.4. - Assigning IP Address

The required IP address can then be either entered, or a recently used IP address can be selected by clicking on an item in the Recent List.

If the “Enable Static” checkbox is checked, then the IP address will be set to static after the IP assignment, thereby disabling future DHCP requests.

Once the IP address window has been accepted, the DHCP server will automatically assign the IP address to the module and then read the Identity object Product name from the device.

The successful assignment of the IP address by the device is indicated by the green background of the associated row.

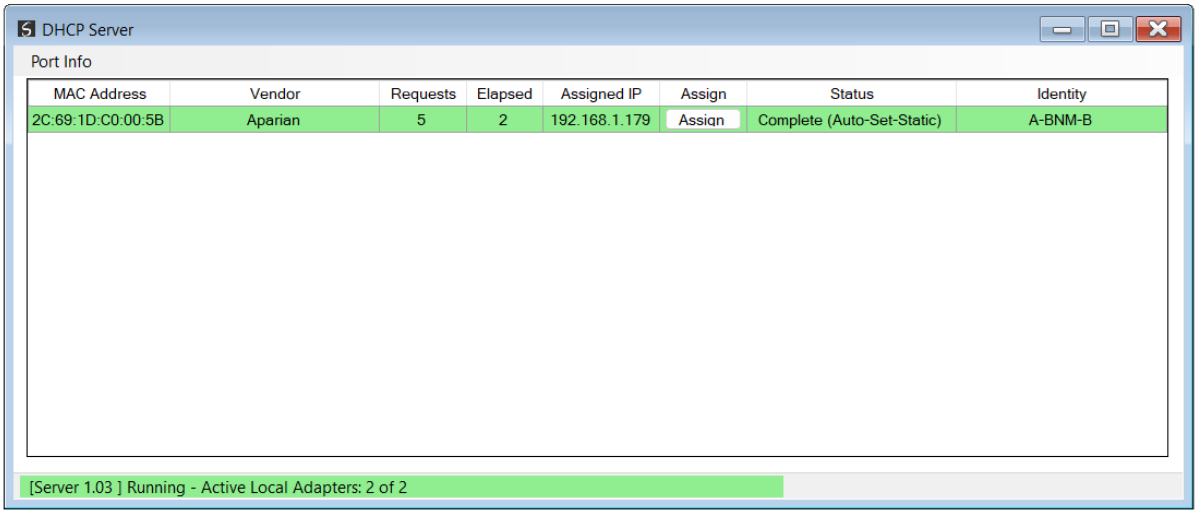


Figure 3.5. - Successful IP address assignment

It is possible to force the module back into DHCP mode by powering up the device with DIP switch 2 set to the On position.

A new IP address can then be assigned by repeating the previous steps.

NOTE: It is important to return DIP switch 2 back to Off position, to avoid the module returning to a DHCP mode after the power is cycled again.

If the module's DIP switch 2 is in the On position during the address assignment, the user will be warned by the following message.

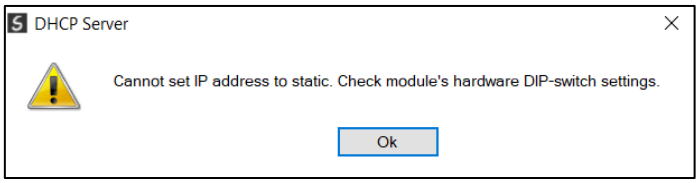


Figure 3.6. - Force DHCP warning

In addition to the setting the IP address, a number of other network parameters can be set during the DHCP process. These settings can be viewed and edited in Slate's Application Settings, in the DHCP Server tab.

Once the DHCP process has been completed, the network settings can be set using the Ethernet Port Configuration via the ***Target Browser***.

The ***Target Browser*** can be accessed under the ***Tools*** menu.

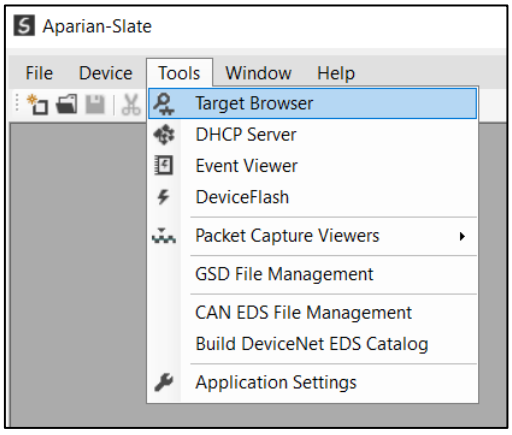


Figure 3.7. - Selecting the Target Browser

The ***Target Browser*** automatically scans the Ethernet network for EtherNet/IP devices.

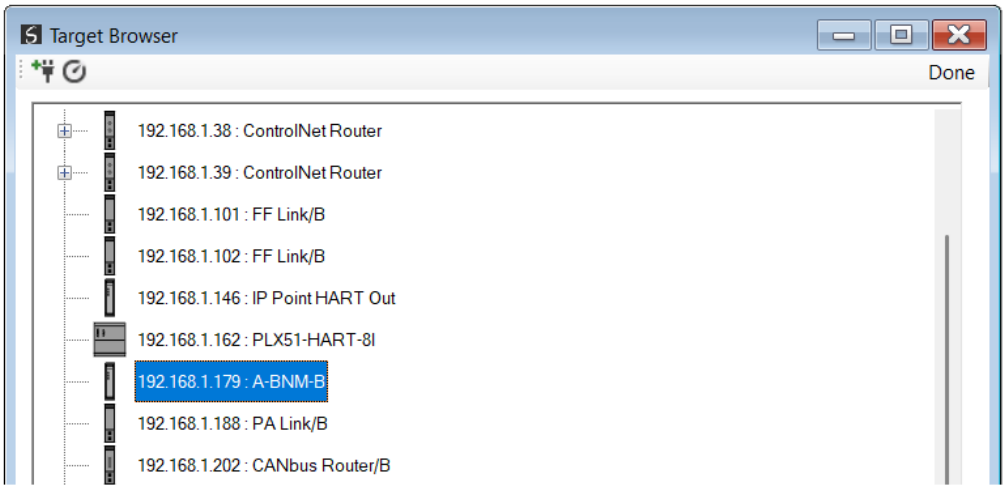


Figure 3.8. - Target Browser

Right-clicking on a device, reveals the context menu, including the ***Port Configuration*** option.

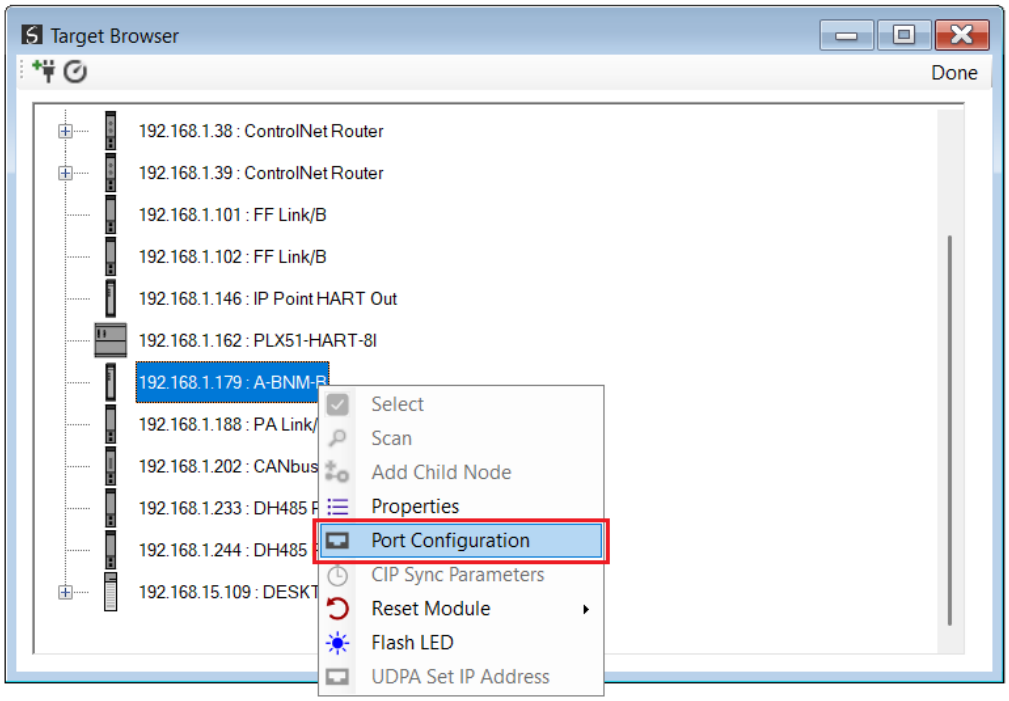


Figure 3.9. - Selecting Port Configuration

All the relevant Ethernet port configuration parameters can be modified using the ***Port Configuration*** window.

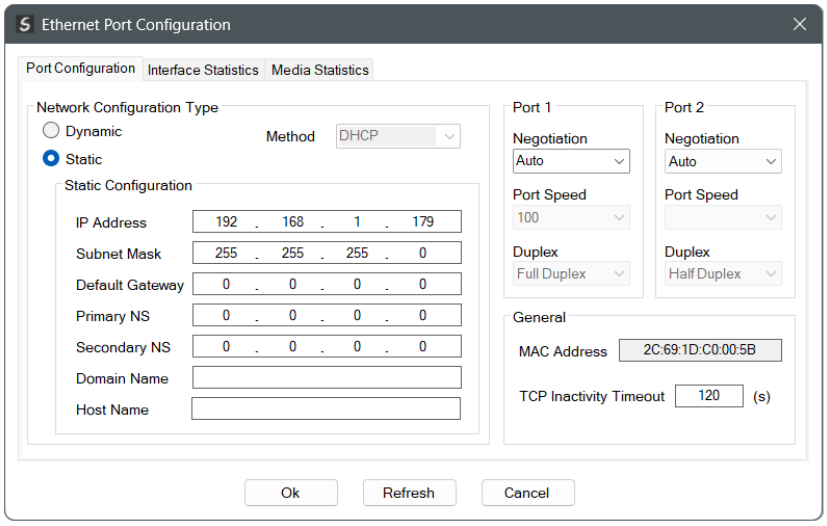


Figure 3.10. - Port Configuration

Alternatively, these parameters can be modified using Rockwell Automation's RSLinx software.

3.3. CREATING A NEW PROJECT

Before the user can configure the module, a new Slate project must be created. Under the ***File*** menu, select ***New***.

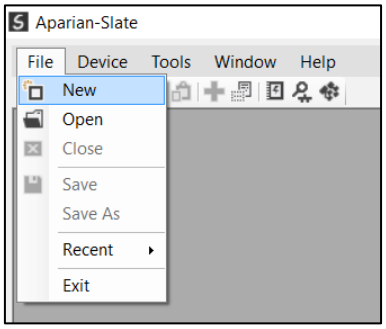


Figure 3.11. - Creating a new project

A Slate project will be created, showing the Project Explorer tree view. To save the project use the ***Save*** option under the ***File*** menu.

A new device can now be added by selecting ***Add*** under the ***Device*** menu.

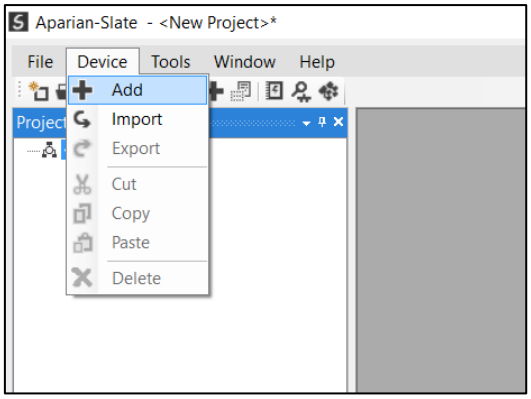


Figure 3.12. - Adding a new device

In the Add New Device window select the BACnet Master and click the Ok button.

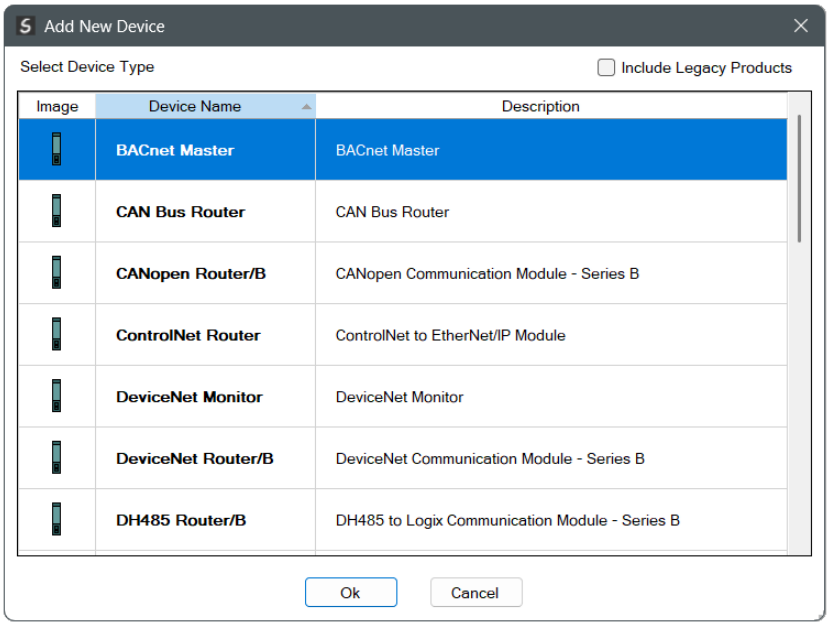


Figure 3.13 – Selecting a new BACnet Master

The device will appear in the Project Explorer tree as shown below, and its configuration window opened. The device configuration window can be reopened by either double clicking the module in the Project Explorer tree or right-clicking the module and selecting ***Configuration***.

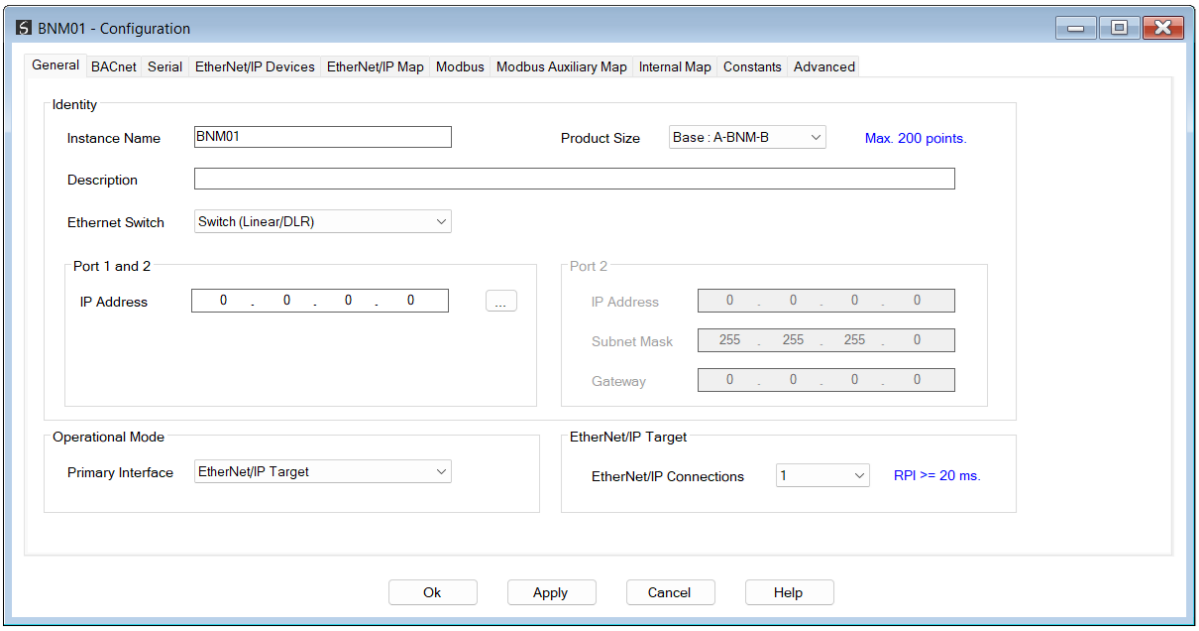


Figure 3.14. – BACnet Master configuration

Refer to the additional information section in this document for Slate's installation and operation documentation.

3.4. GENERAL PARAMETERS

The BACnet parameters will be configured by Slate. When downloading this configuration into the module it will be saved in non-volatile memory that persists when the module is powered down.

NOTE: When a firmware upgrade is performed the module will clear all the module configuration stored in non-volatile storage.

The general configuration is shown in the figure below. The BACnet Master general configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting ***Configuration***.

NOTE: When browsing for the module in the *General* configuration window, the user will need to first apply the *Product Size* parameter to the offline configuration before the module will be recognized (and highlights in blue) in the target browser.

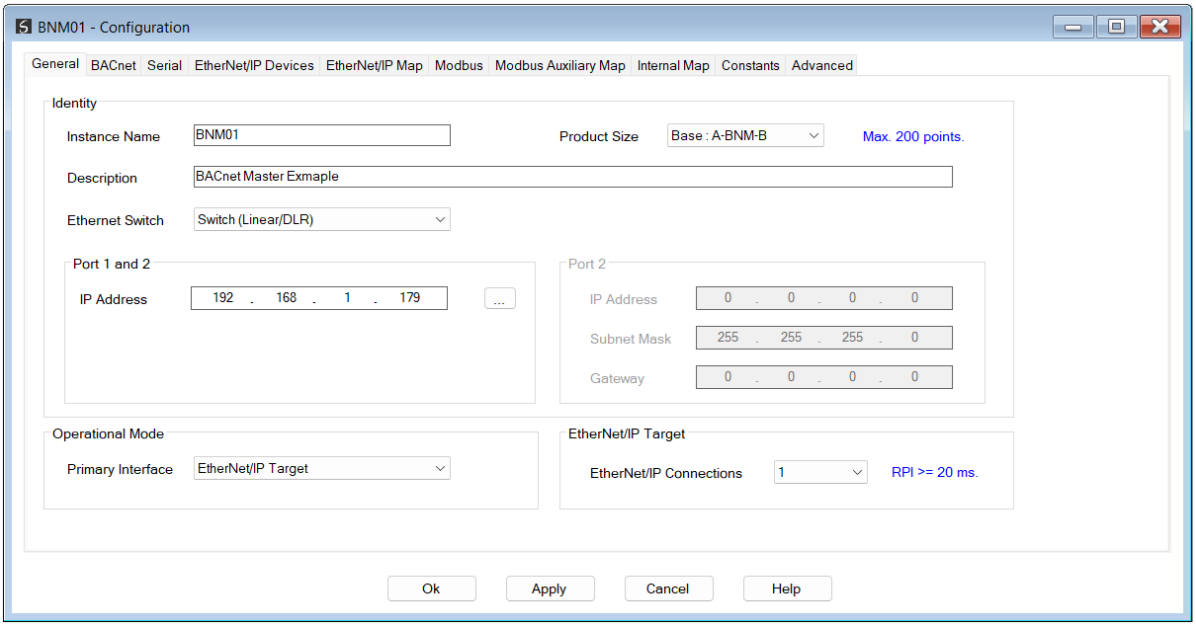


Figure 3.15 - General Configuration

The general configuration consists of the following parameters:

Parameter	Description
Instance Name	This parameter is a user defined name to identify between various BACnet Masters.
Product Size	Select the specific BACnet Product • Base (A-BNM-B) – 200 data point and 50 BACnet device limit • Pro (A-BNM-P) – 1000 data point and 250 BACnet device limit • Max (A-BNM-M) – 2500 data point and 250 BACnet device limit **NOTE**: The user will not be able to download the configuration to the module if the Product Size does not match the product that has been purchased.
Description	This parameter is used to provide a more detailed description of the application for the module.
Ethernet Switch	The operational mode of the embedded Ethernet switch. **Switch (Linear/DLR)** This is the default mode where the embedded switch is enabled. The two Ethernet ports are identical in functionality and (foreign) traffic received on one port is transmitted out the other. Both ports use the IP address configured using DHCP or set the Target browser. **Split (Dual-IP)** In this mode the embedded switch is disabled. Traffic received on one port is not transmitted to the other.

	Port 1: The IP address is the one configured using DHCP or via the Target browser. EtherNet/IP requests and connections will be processed. Modbus TCP requests will be processed. Port 2: The IP address is the one contained within the configuration below (Port 2). EtherNet/IP requests and connections will not be processed. Modbus TCP requests will be processed.
Port 1 IP Address	The primary configured IP address of the module.
Port 2 IP Address	**THIS IS RELEVANT ONLY IN SPLIT (DUAL-IP) ETHERNET SWITCH MODE** The IP address used on Port 2.
Port 2 Subnet Mask	**THIS IS RELEVANT ONLY IN SPLIT (DUAL-IP) ETHERNET SWITCH MODE** The Subnet Mask used on Port 2.
Port 2 Gateway	**THIS IS RELEVANT ONLY IN SPLIT (DUAL-IP) ETHERNET SWITCH MODE** The Default Gateway address used on Port 2.  **NOTE**: If the IP address ranges for Port 1 and Port 2 overlap, then only one Port's Gateway address can be non-zero.
Primary Interface	**EtherNet/IP Target** A Logix controller can own the BACnet Master over EtherNet/IP using 1-25 class 1 connections. **Modbus Server** A Modbus Client can read and write data to the module which can then be mapped to one or more BACnet devices. The module can operate as a Modbus Server on Ethernet TCP, RTU-232 and RTU-485. **Modbus Client** A module can read and write data from various Modbus devices which can then be mapped to one or more BACnet devices. The module can operate as a Modbus Client on Ethernet TCP, RTU-232 and RTU-485. **EtherNet/IP Originator** As an EtherNet/IP originator, the module can use two methods to read and write data to and from an EtherNet/IP device (IO): *EtherNet/IP Class 1 Connection* The BACnet Master can own EtherNet/IP IO by using the Slate software to configure the IO connections. *EtherNet/IP Explicit Messaging* The BACnet Master can exchange data with up to 10 EtherNet/IP devices using explicit messaging.
EtherNet/IP Connections	The number of connections (1-25) the module will use when operating as an EtherNet/IP target.

	NOTE: This value must match the connection count configured in the Logix IO tree. (EDS AOP).

Table 3.1 - General configuration parameters

3.5. BACNET CONFIGURATION

The BACnet Communication configuration window is opened by either double clicking on the module in the tree, or right-clicking the module and selecting ***Configuration*** and then selecting the ***BACnet*** tab.

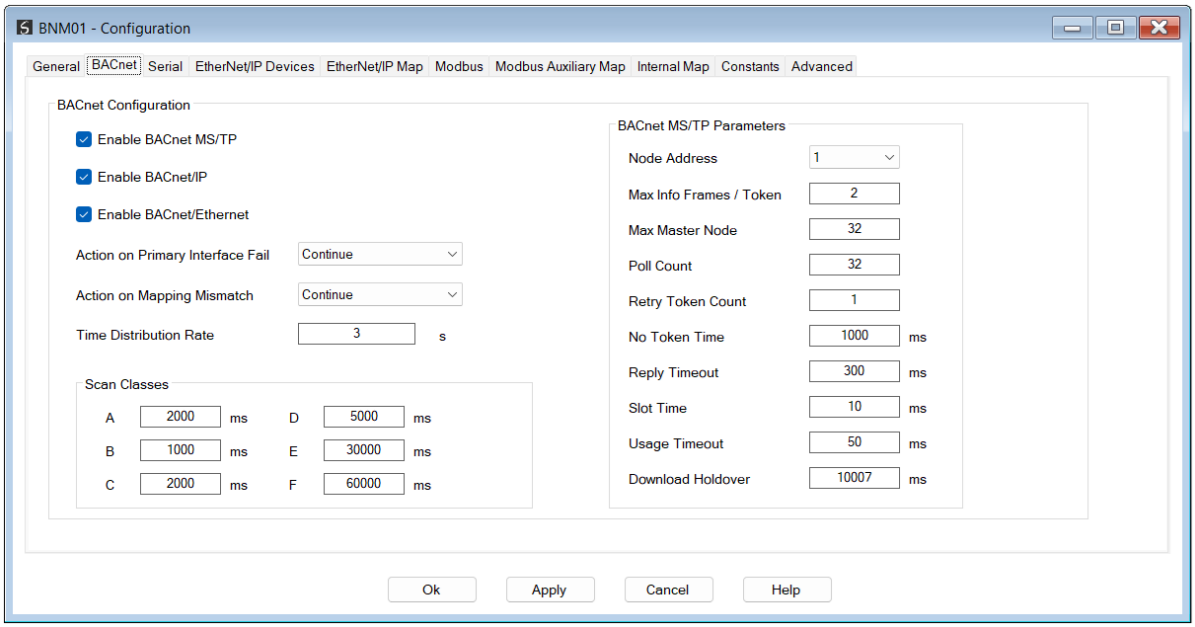


Figure 3.16 – BACnet Communication Configuration

The **BACnet Communication** configuration consists of the following parameters:

Parameter	Description
BACnet Configuration	
Enable BACnet MS/TP	Enable the BACnet MS/TP network (RS485). **NOTE:** Modbus RTU485 cannot be selected if BACnet MS/TP has been enabled.
Enable BACnet/IP	Enable the BACnet/IP network
Enable BACnet/Ethernet	Enable the BACnet/Ethernet network

Action on Primary Interface Fail	The action taken by the BACnet Master when the Primary Interface (EtherNet/IP / Modbus) fails. **Stop** BACnet communication will cease and the BACnet devices will go Offline. **Zero Data** BACnet communication will continue, but the "output" values written to BACnet devices will be set to zero. **Continue** BACnet communication will continue as normal, with the last known "output" values written to the BACnet devices.
Action on Mapping Mismatch	The action taken by the BACnet Master when the EtherNet/IP controller (e.g. Logix) has a mapping checksum different from that in the BACnet master. **Stop** BACnet communication will cease and the BACnet devices will go Offline. **Continue** BACnet communication will continue as normal.
Time Distribution Rate	The rate (in seconds) at which the BACnet Master distributes network time. A value of 0s disables the time distribution function.
Scan Classes	The time period (milliseconds) for each of the 6 (A…F) scan classes. These classes are referenced in the BACnet device configuration to determined how often a BACnet property is polled.
BACnet MS/TP Parameters	
Node Address	The local node address of the BACnet Master.
Max Info Frames Per Token	The maximum number of information frames a node may send before it must pass the token. (N-MAX_INFO_FRAMES)
Max Master Node	Specifies the highest allowable address for master nodes. (1-127). (N-MAX_MASTER)
Poll Count	The number of tokens received or used before a Poll For Master cycle is executed. (N-POLL_COUNT)
Retry Token Count	The number of retries on sending the Token. (N-RETRY_TOKEN)
No Token Time	The time (milliseconds) without a DataAvailable or ReceiveError event before declaration of a loss of token. (T-NO_TOKEN)
Reply Timeout	The minimum time (milliseconds) without a DataAvailable or ReceiveError event that a node must wait for a station to begin replying to a confirmed request. (T-REPLY_TIMEOUT)

Slot Time	The width of the time slot (milliseconds) within which a node may generate a token. (T-SLOT)
Usage Timeout	The minimum time (milliseconds) without a DataAvailable or ReceiveError event that a node must wait for a remote node to begin using a token or replying to a Poll For Master frame. (T-USAGE_TIMEOUT)
Download Holdover	The time (milliseconds) that the MS/TP communication will remain active during a module download event.

Table 3.2 - BACnet Communication configuration parameters

3.6. SERIAL CONFIGURATION

The Serial port configuration window is opened by either double clicking on the module in the tree, or right-clicking the module and selecting ***Configuration*** and then selecting the ***Serial*** tab.

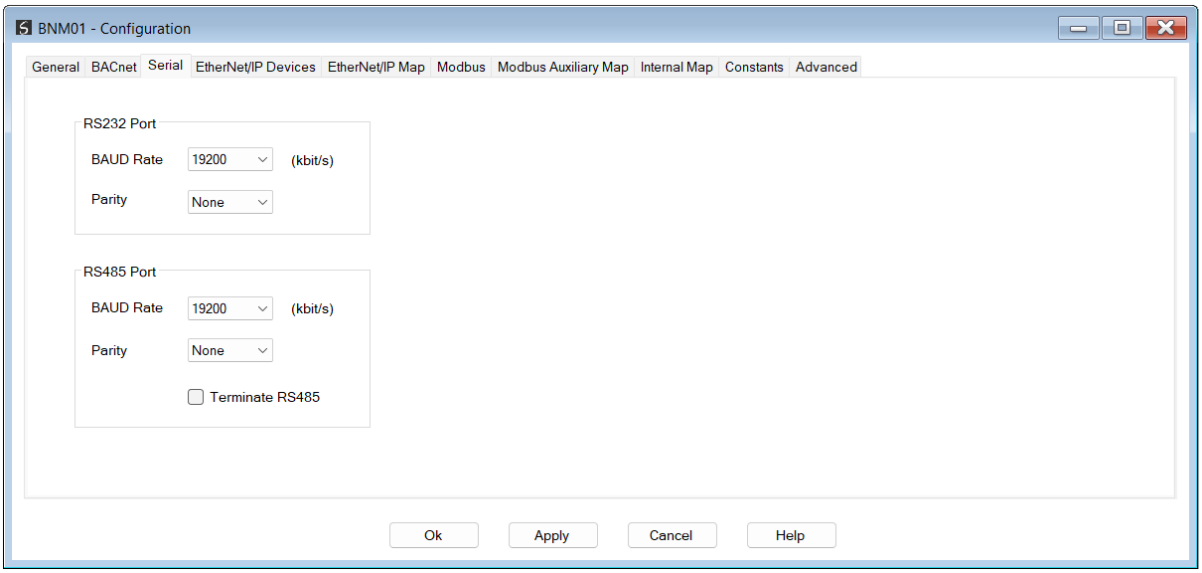


Figure 3.17 – Serial Port Configuration

The **Serial Port** configuration consists of the following parameters:

Parameter	Description
RS232 Port	
BAUD Rate	The BAUD rate specifies the speed at which the data is transferred across the RS232 serial network. The module provides the following speeds: • 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200
Parity	The parity parameter will configure the parity of the RS232 serial port. The module allows for **Even**, **Odd**, or **None** parity setting.

RS485 Port	
BAUD Rate	The BAUD rate specifies the speed at which the data is transferred across the RS485 serial network. The module provides the following speeds: • 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200
Parity	The parity parameter will configure the parity of the RS485 serial port. The module allows for **Even**, **Odd**, or **None** parity setting.
Termination	Enables the on-board 125Ω RS485 terminating resistor.

Table 3.3 – Serial port parameters

3.7. PRIMARY INTERFACE

The BACnet Master module supports four different modes for the Primary Interface.

3.7.1. ETHERNET/IP TARGET

A controller (e.g. Logix controller) can own the BACnet Master over EtherNet/IP using 1 to 25 Class 1 EtherNet/IP connections when the Primary Interface is set to EtherNet/IP target. This will allow the BACnet Master to exchange data with the controller using the input and output assembly of the Class 1 EtherNet/IP connections. Data from the BACnet devices can be mapped to the Logix controller over EtherNet/IP.

The user will need to add the BACnet Master to the Logix IO tree under an EtherNet/IP bridge (e.g. 1756-EN2TR) or Ethernet Logix controller (e.g. 1756-L85E).

3.7.1.1. STUDIO / LOGIX 5000 CONFIGURATION

A. ADD MODULE TO ETHERNET/IP I/O CONFIGURATION

The BACnet Master can be easily integrated with Allen-Bradley Logix family of controllers. Integration with the Logix family in Studio5000 makes use of the EDS Add-On-Profile (AOP).

NOTE: Logix version 21 and newer supports EDS AOPs.

Before the module can be added to the tree the module's EDS file must be registered.

Using RSLinx, the EDS file can be uploaded from the device after which the ***EDS Hardware Installation*** tool will be invoked to complete the registration.

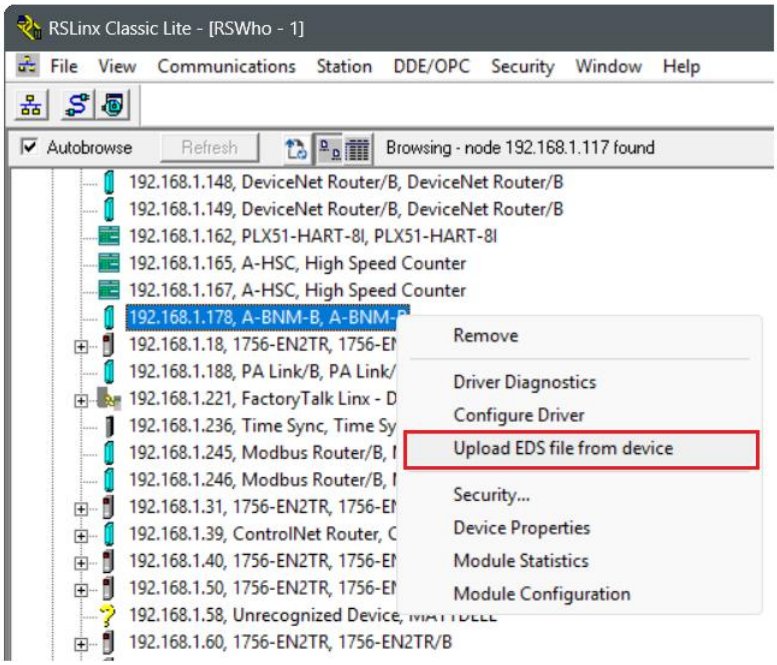


Figure 3.18 – EDS file upload from BACnet Master

Alternatively, the EDS file can be downloaded from the product web page at www.aparian.com and registered manually using the ***EDS Hardware Installation Tool*** shortcut under the ***Tools*** menu in Studio 5000.

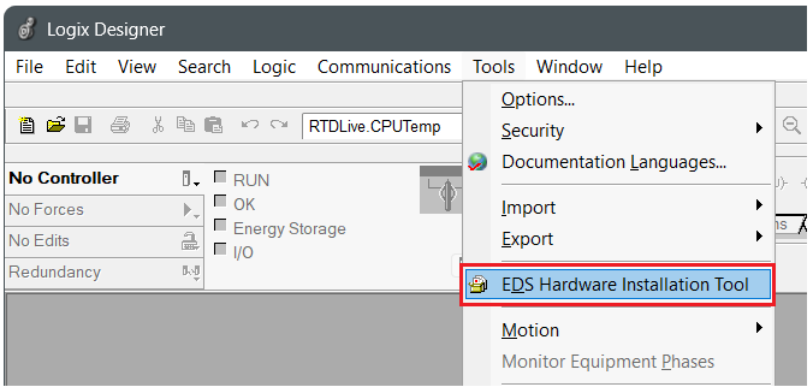


Figure 3.19 - EDS Hardware Installation Utility

After the EDS file has been registered, the module can be added to the Logix IO tree in Studio 5000. Under a suitable Ethernet bridge module in the tree, select the Ethernet network, right-click and select the ***New Module*** option.

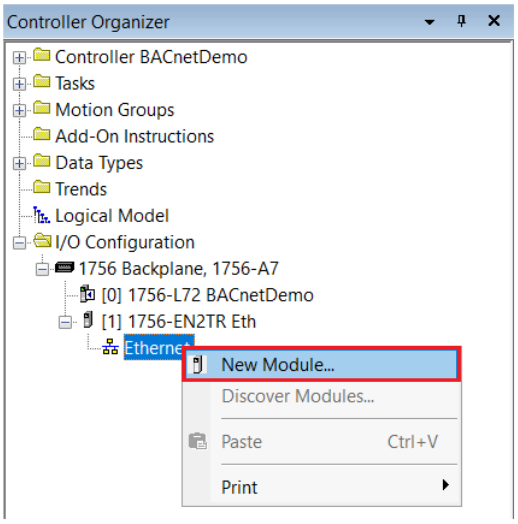


Figure 3.20 – Adding a module

The module selection dialog will open. To find the module more easily, use the Vendor filter to select only the Aparian modules as shown in the figure below.

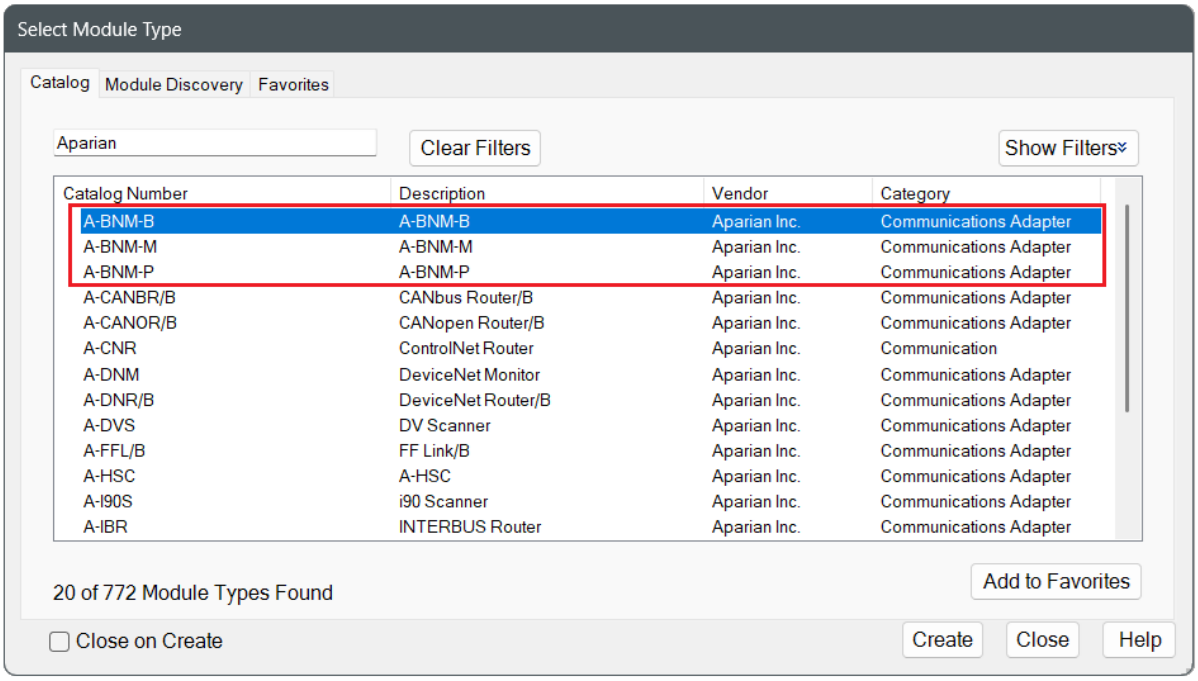


Figure 3.21 – Selecting the module

Locate and select one of the BACnet Master module variants (A-BNM-B, A-BNM-M, or A-BNM-P) and select the ***Create*** option. The module configuration dialog will open, where the user must specify the Name and IP address as a minimum to complete the instantiation.

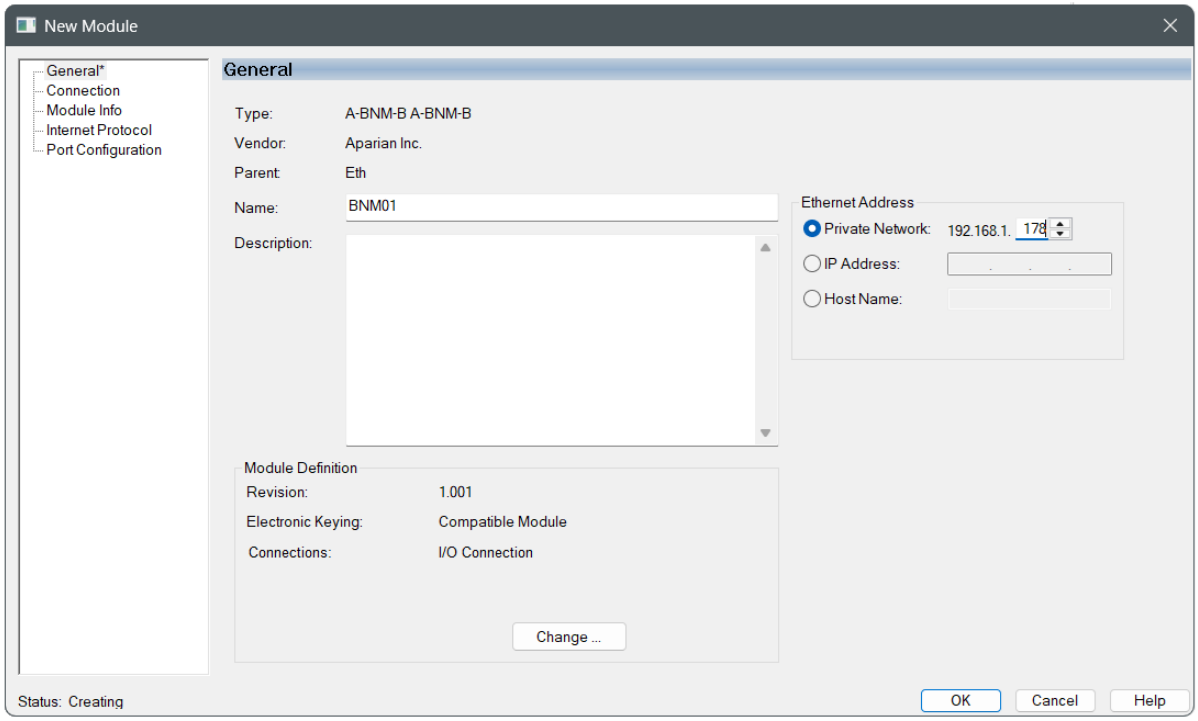


Figure 3.22 – Module instantiation

The BACnet Master supports up to 25 class 1 EtherNet/IP connections. The user will need to **ensure that the number of connections configured in the General tab of the module configuration (Slate) matches the selected connection count in Logix (Studio 5000)**.

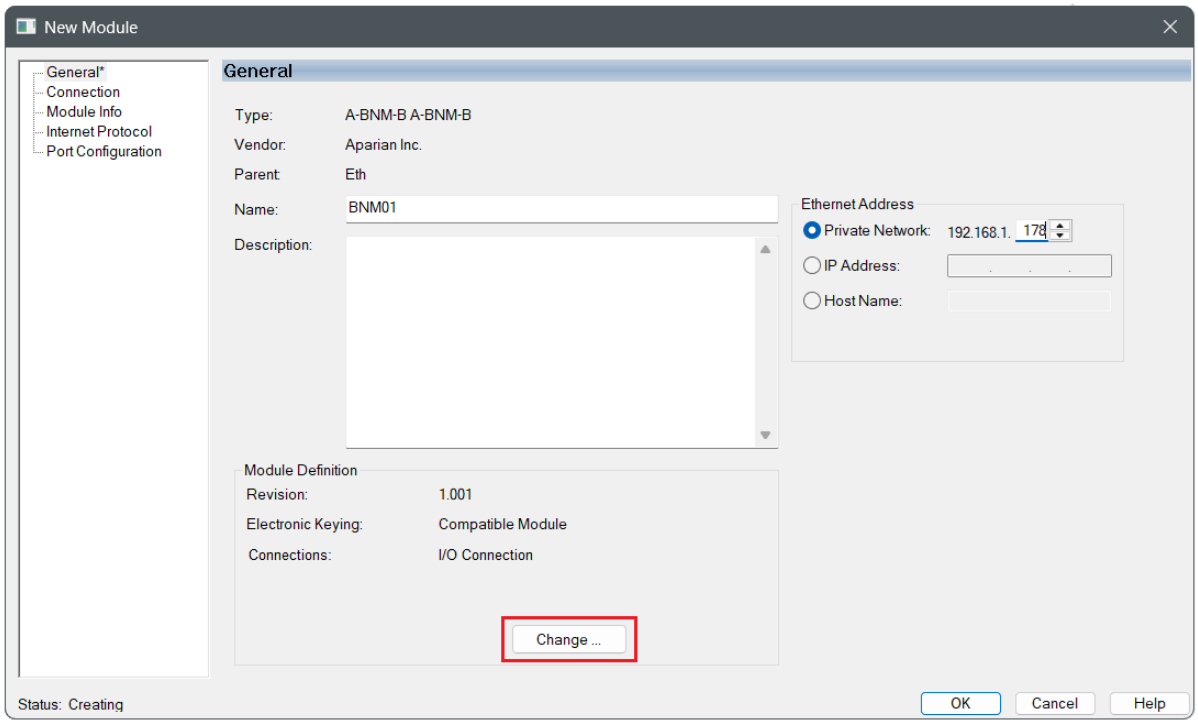


Figure 3.23 – Change number of IO Connections

Next the user will need to select the number of connections required.

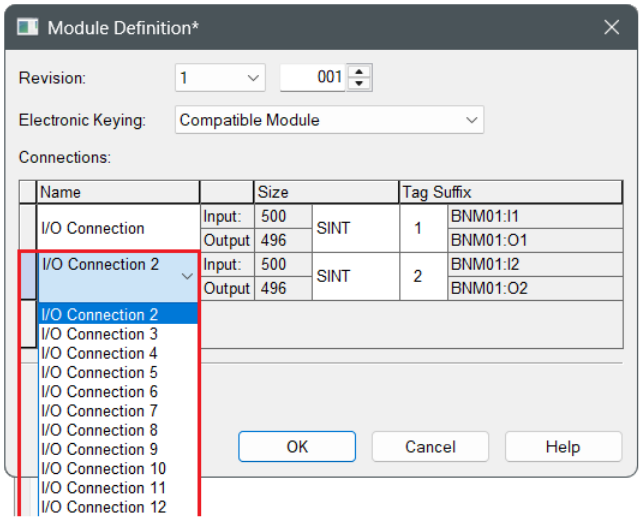


Figure 3.24 – Selection of IO Connections

The minimum allowed RPI (Requested Packet Interval) depends on the number of configured connections, as specified in the table below:

Connection Count	Minimum RPI
1 – 5	20 ms
6 – 10	40 ms
11 – 15	60 ms
16 – 20	80 ms
21 – 25	100 ms

Table 3.3 – Minimum allowed RPI values

Now the BACnet Master module will be in the Logix IO tree.

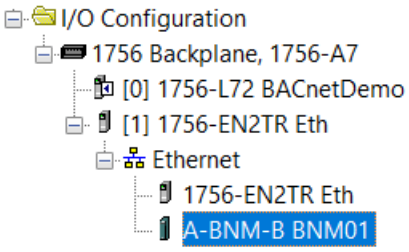


Figure 3.25 – Logix IO tree

The Module Defined Data Types will automatically be created during the instantiation process.

B. LOGIX MAPPING

Slate will generate the required UDTs (User-defined Data Types) and Routines (based on the configured Internal Map) to map the required BACnet data. The user will need to select the ***Recommend*** button in the Internal Mapping to auto-populate the Internal Mapping which can then be used to generate the L5X file for Logix mapping, routines, and UDTs

NOTE: If the ***Enable Auto Internal Mapping*** option (Advanced tab) has been enabled then the ***Recommend*** button and Mapping grid will be disabled, because it will automatically be applied.

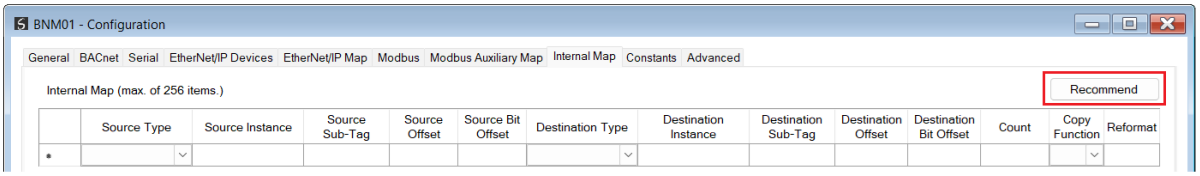


Figure 3.26 – Internal Mapping Recommend

	Source Type	Source Instance	Source Sub-Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Sub-Tag	Destination Offset	Destination Bit Offset	Count	Copy Function	Reformat
1	System		Status	0		EIP Target	Connection 0		0		96	Byte to Byte	None
2	EIP Target	Connection 0		0		System		Control	0		4	Byte to Byte	None
3	BACnet Device	Dev1002	Status	0		EIP Target	Connection 0		96		4	Byte to Byte	None
4	BACnet Device	Dev1002	Data	0		EIP Target	Connection 0		100		32	Byte to Byte	None
5	BACnet Device	SimDevice_1...	Status	0		EIP Target	Connection 0		132		4	Byte to Byte	None
6	BACnet Device	SimDevice_1...	Data	0		EIP Target	Connection 1		0		480	Byte to Byte	None
7	EIP Target	Connection 1		0		BACnet Device	SimDevice_1...	Data	0		40	Byte to Byte	None
8	BACnet Device	ETH_Dev01	Status	0		EIP Target	Connection 1		480		4	Byte to Byte	None
9	BACnet Device	ETH_Dev01	Data	0		EIP Target	Connection 2		0		52	Byte to Byte	None
10	EIP Target	Connection 2		0		BACnet Device	ETH_Dev01	Data	0		8	Byte to Byte	None
11	BACnet Device	BMTDIO42_04	Status	0		EIP Target	Connection 2		52		4	Byte to Byte	None
12	BACnet Device	BMTDIO42_04	Data	0		EIP Target	Connection 2		56		12	Byte to Byte	None
13	EIP Target	Connection 2		8		BACnet Device	BMTDIO42_04	Data	0		8	Byte to Byte	None

Figure 3.27 – Internal Mapping Auto Populated

The user can then generate the required Logix and UDTs by right-clicking on the module in Slate and selecting the ***Generate Logix L5X*** option.

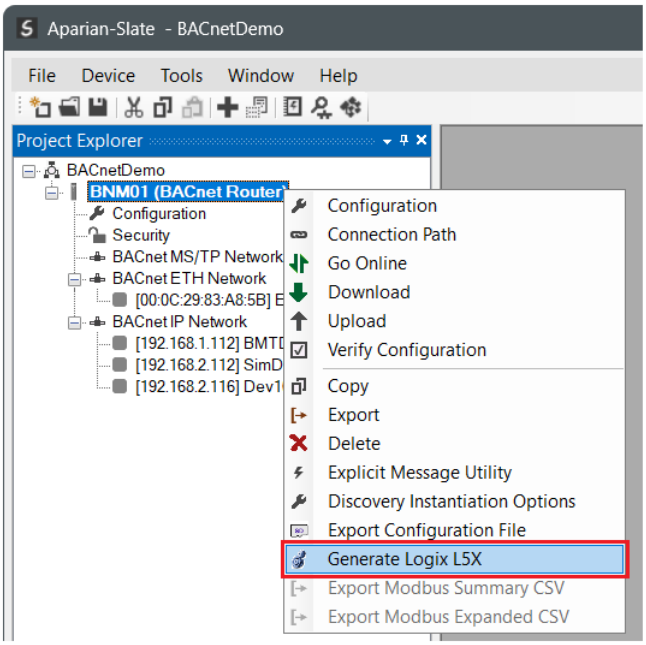


Figure 3.28 – Selecting Generate Logix L5X

The user will then be prompted to select a suitable file name and path for the L5X file.

This L5X file can now be imported into the Studio 5000 project by right-clicking on a suitable ***Program*** and selecting ***Add***, and then ***Import Routine***.

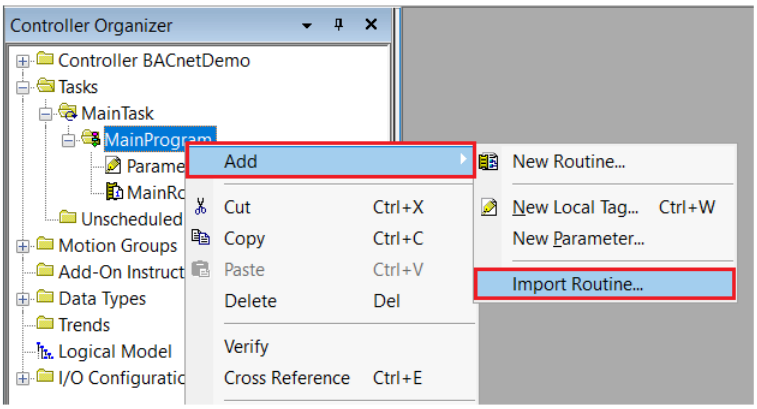


Figure 3.29 – Importing the L5X file into Studio 5000

In the file open dialog select the previously created L5X file and accept the import by pressing ***Ok***.

NOTE: When importing the L5X at a subsequent time, it is important to set the Studio 5000 import options to "**Overwrite**" for both the ***Tags*** and ***Data Type*** sections, otherwise the updated structures will be ignored.

NOTE: The user can select the *Recommend* button in the Internal Map to auto map the BACnet status and data to recommended Modbus Registers.

After selecting the Import Routine action, select the ***Tags*** section on the left, then right-click on the ***Operation*** heading, and then select the "***Change …. To Overwrite***" option.

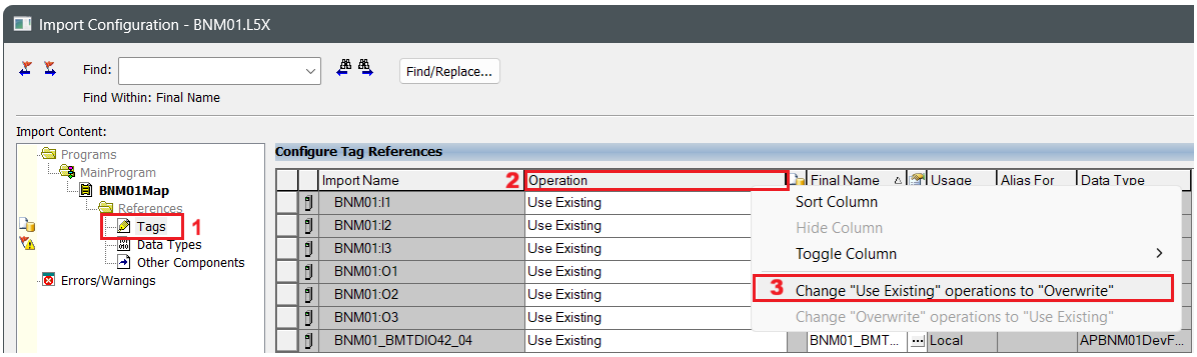


Figure 3.30 – Studio 5000 Import – Set Tags to Overwrite

Then select the ***Data Types*** section on the left, then right-click on the ***Operation*** heading, and then select the "***Change …. To Overwrite***" option.

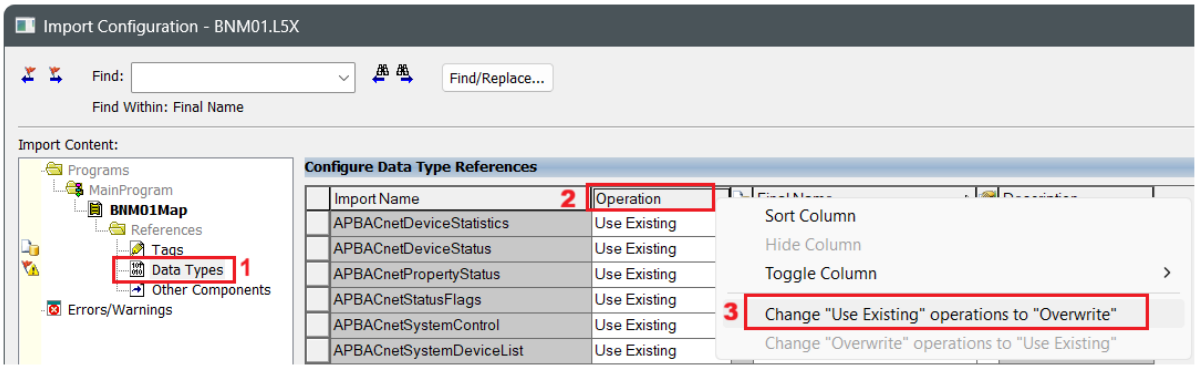


Figure 3.31 – Studio 5000 Import – Set Data Types to Overwrite

The import will create the following:

- Mapping Routine
- Multiple UDT (User-Defined Data Types)
- Multiple Controller Tags

NOTE: Since the imported mapping routine is not a Main Routine, it will need to be called from the current Main Routine.

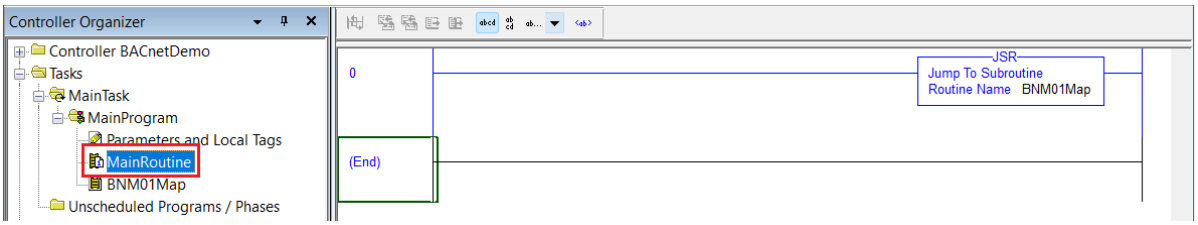


Figure 3.32 – Calling the mapping routine

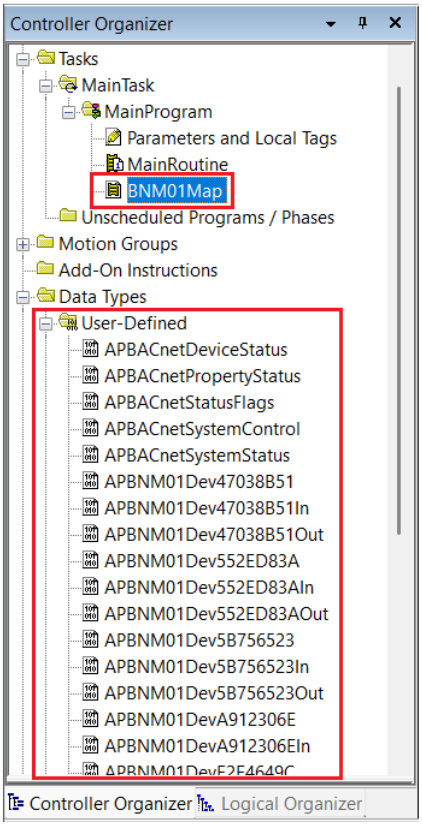


Figure 3.33 – Imported Logix Objects

BACnet Device specific UDTs and tags are created for each BACnet Device in the configuration.

If the ***System Status*** option was selected during the mapping, then the BACnet Master System Status structure will be included in the Input tag.

The System tag displays the status and diagnostics information of the BACnet Master.

Name	Value	Style	Data Type	Description
BNM01_SystemStatus	{...}		APBACnetSystemStatus	
BNM01_SystemStatus.ConfigValid	1	Decimal	BOOL	Configuration Valid
BNM01_SystemStatus.EIPOriginatorCommsOk	0	Decimal	BOOL	EtherNet/IP Originator 0=Fail, 1=Ok
BNM01_SystemStatus.ModbusOnline	0	Decimal	BOOL	0=Offline, 1=Online
BNM01_SystemStatus.EIPOwned	1	Decimal	BOOL	EtherNet/IP Target: 0=Not-Owned, 1=Owned
BNM01_SystemStatus.ControllerRun	1	Decimal	BOOL	Controller Mode: 0=Program, 1=Run
BNM01_SystemStatus.NTPOk	1	Decimal	BOOL	NTP Status: 0=Fail, 1=Ok
BNM01_SystemStatus.BACNetMSTPOk	0	Decimal	BOOL	All Device Status: 0=Fault, 1=Ok
BNM01_SystemStatus.BACNetIPOk	1	Decimal	BOOL	All Device Status: 0=Fault, 1=Ok
BNM01_SystemStatus.BACNetETHOk	1	Decimal	BOOL	All Device Status: 0=Fault, 1=Ok
BNM01_SystemStatus.MappignCRCMatch	1	Decimal	BOOL	Mapping CRC: 0=Mismtach, 1=Match
BNM01_SystemStatus.ConfigCRC	16#1c7c	Hex	INT	Configuration Checksum
BNM01_SystemStatus.TransactionRate	5	Decimal	INT	Transaction Rate
BNM01_SystemStatus.MSTPDevicesOnline	1	Decimal	INT	Number of BACnet MSTP Devices Online
BNM01_SystemStatus.ETHDevicesOnline	0	Decimal	INT	Number of BACnet Ethernet Devices Online
BNM01_SystemStatus.IPDevicesOnline	3	Decimal	INT	Number of BACnet IP Devices Online
BNM01_SystemStatus.Reserved1	0	Decimal	INT	Reserved1
BNM01_SystemStatus.MSTPDevicesExchangeCount	0	Decimal	INT	BACNet MS/TP Exchange Count
BNM01_SystemStatus.ETHDevicesExchangeCount	124	Decimal	INT	BACNet Ethernet Exchange Count
BNM01_SystemStatus.IPDevicesExchangeCount	365	Decimal	INT	BACNet IP Exchange Count
BNM01_SystemStatus.Reserved2	0	Decimal	INT	Reserved2
BNM01_SystemStatus.DeviceOnline	{...}	Decimal	BOOL[256]	Devices Online
BNM01_SystemStatus.DeviceDataExchange	{...}	Decimal	BOOL[256]	Devices Data Exchange
BNM01_SystemStatus.ActualMappingCRC	16#efe7	Hex	INT	ActualMappingCRC
BNM01_SystemStatus.ReceivedMappingCRC	16#efe7	Hex	INT	ReceivedMappingCRC

Figure 3.34 – System Status tag

3.7.1.2. INTERNAL DATA SPACE MAPPING

When the module is operating as an EtherNet/IP Target, the data from the originator device (e.g. Logix Controller) can be mapped to the BACnet interface using the Internal Map. The Internal Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the *Internal Map* tab.

A. IDS COPY – ETHERNET/IP TARGET SOURCE

When copying data from a connection originator (e.g. the output assembly from the Logix Controller) to the BACnet interface, the source type needs to be EIP Target.

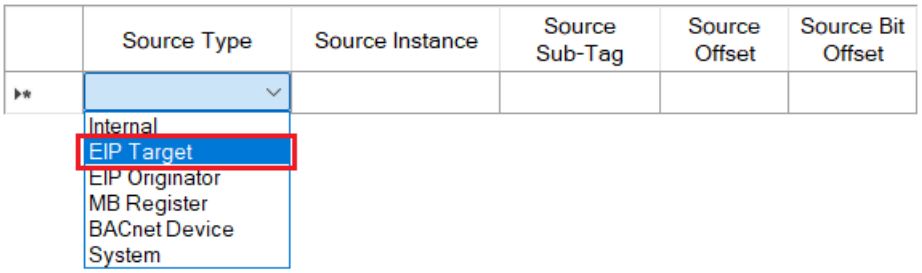


Figure 3.35 – IDS Copy – EtherNet/IP Target Source Type

The ***Source Instance*** will be the connection number, which can be Connection 0 to 24, based on the number of connections configured. The ***Source Offset*** is the offset in the EtherNet/IP output assembly from where the data must be copied. The ***Count*** is the number of **bytes** that will be copied.

See the Internal Data Space Mapping section for more information regarding the operation.

B. IDS COPY – ETHERNET/IP TARGET DESTINATION

When copying data from the BACnet interface to the EtherNet/IP Target input assembly, the destination type needs to be EIP Target.

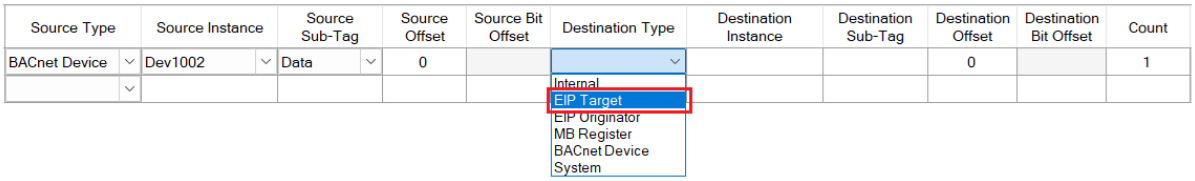


Figure 3.36 – IDS Copy – EtherNet/IP Target Destination Type

The ***Destination Instance*** will be the connection number, which can be Connection 0 to 24, based on the number of connections configured. The ***Destination Offset*** is the offset of the EtherNet/IP input assembly to where the data must be copied. The ***Count*** is the number of **bytes** that will be copied.

3.7.2. MODBUS (SERVER AND CLIENT)

The BACnet Master can operate as a Modbus Server or Modbus Client for Modbus TCP, Modbus RTU-232 and Modbus RTU-485.

When configured as a Modbus Server, a remote Modbus Client can read and write to the Modbus Register range (0-65535) in the BACnet Master.

When configured as a Modbus Client, the user will need to configure the Modbus Auxiliary Map. This map will allow the user to configure various read and write functions to external Modbus Registers, to and from the internal Modbus registers.

The Modbus Parameters will need to be configured for either Modbus modes.

The Modbus configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the ***Modbus*** tab.

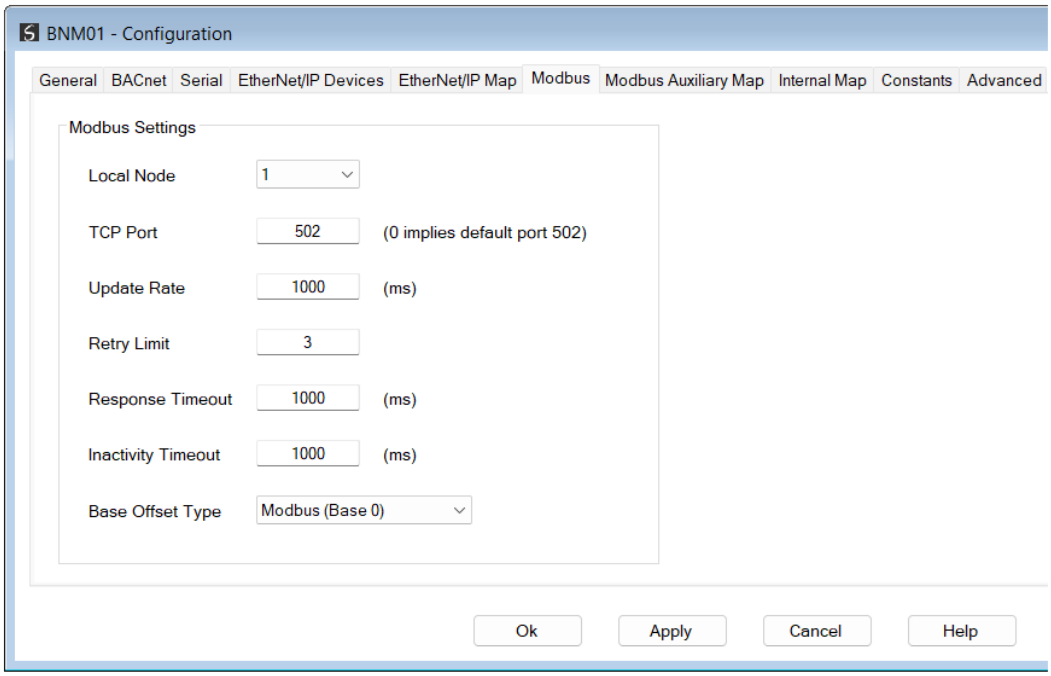


Figure 3.37 – Modbus Configuration

The Modbus Communication configuration consists of the following parameters:

Parameter	Description
Local Node	The Modbus Node address assigned to the BACnet Master. **(Modbus Client mode only)**
TCP Port	The TCP port to be used for the Modbus-TCP communication. By default, the module will use the standard TCP port 502.
Update Rate	The period (in milliseconds) between Client requests to the target Modbus device. **(Modbus Client mode only)**
Retry Limit	The number of successive Modbus request retries, before the request is deemed to have failed. **(Modbus Client mode only)**
Response Timeout	The time (in milliseconds) the module will wait for a valid Modbus response. **(Modbus Client mode only)**
Inactivity Timeout	The amount of time during which no Modbus requests have been received before the BACnet Master indicates that the connection to the Modbus Client is no longer active. **(Modbus Server mode only)**
Base Offset Type	**Modbus (Base 0)** Conventional Modbus addressing where the first address is 0. **PLC (Base 1)**

	PLC addressing, where the first address is 1.

Table 3.3 – Modbus parameters

3.7.2.1. INTERNAL DATA SPACE MAPPING

When the module is operating as a Modbus Server or Client, the internal Modbus Registers can be mapped to the BACnet data using the Internal Map. The Internal Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the *Internal Map* tab.

NOTE: The user can select the *Recommend* button in the Internal Map to auto map the BACnet status and data to recommended Modbus Registers.

NOTE: If the ***Enable Auto Internal Mapping*** option (Advanced tab) has been enabled then the ***Recommend*** button and Mapping grid will be disabled.

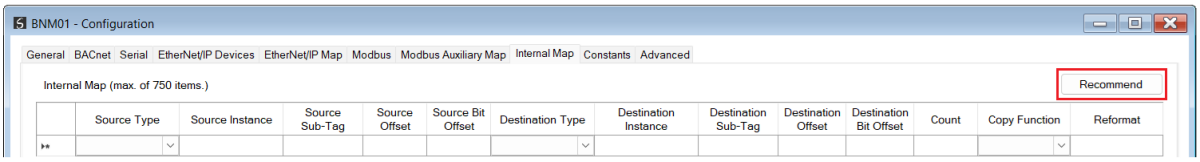


Figure 3.38 – Modbus – Internal Mapping Recommend

	Source Type	Source Instance	Source Sub-Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Sub-Tag	Destination Offset	Destination Bit Offset	Count	Copy Function	Reformat
1	System		Status	0		MB Register	HR		1000		96	Byte to Byte	None
2	MB Register	HR		1048		System		Control	0		4	Byte to Byte	None
3	BACnet Device	Dev1002	Status	0		MB Register	HR		1050		4	Byte to Byte	None
4	BACnet Device	Dev1002	Data	0		MB Register	HR		1052		32	Byte to Byte	None
5	BACnet Device	SimDevice_1...	Status	0		MB Register	HR		1068		4	Byte to Byte	None
6	BACnet Device	SimDevice_1...	Data	0		MB Register	HR		1070		480	Byte to Byte	None
7	MB Register	HR		1310		BACnet Device	SimDevice_1...	Data	0		40	Byte to Byte	None
8	BACnet Device	ETH_Dev01	Status	0		MB Register	HR		1330		4	Byte to Byte	None
9	BACnet Device	ETH_Dev01	Data	0		MB Register	HR		1332		52	Byte to Byte	None
10	MB Register	HR		1358		BACnet Device	ETH_Dev01	Data	0		8	Byte to Byte	None
11	BACnet Device	BMTDIO42_04	Status	0		MB Register	HR		1362		4	Byte to Byte	None
12	BACnet Device	BMTDIO42_04	Data	0		MB Register	HR		1364		12	Byte to Byte	None
13	MB Register	HR		1370		BACnet Device	BMTDIO42_04	Data	0		8	Byte to Byte	None

Figure 3.39 – Modbus – Internal Mapping Updated

A. IDS COPY – MODBUS SOURCE

When copying Modbus data to the BACnet interface, the ***Source Type*** needs to be ***MB Register***.

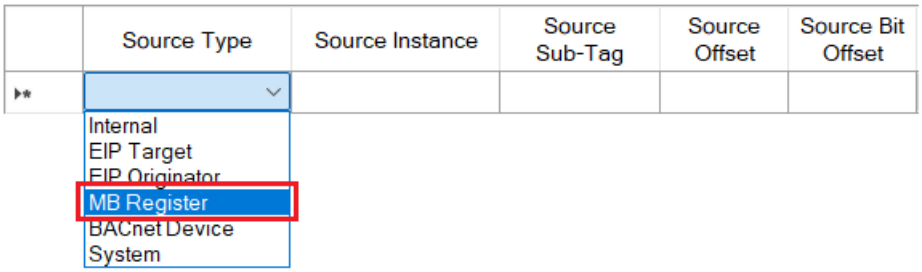


Figure 3.40 – IDS Copy - Modbus Source Type

The ***Source Instance*** will be the Modbus register type required.

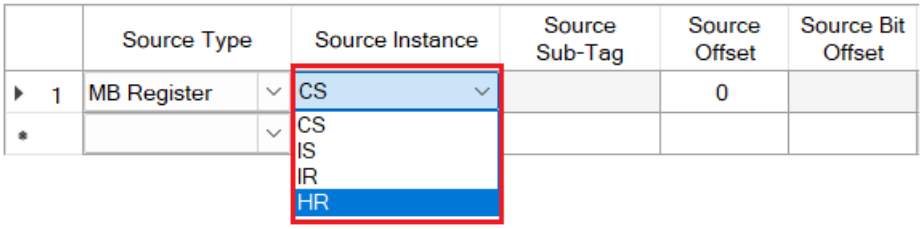


Figure 3.41 – IDS Copy - Modbus Source Instance

The ***Source Offset*** is the Modbus Register offset from where the data must be copied. The ***Count*** is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

B. IDS COPY – MODBUS DESTINATION

When copying data from the BACnet interface to a Modbus Register, the ***Destination Type*** needs to be ***MB Register***.

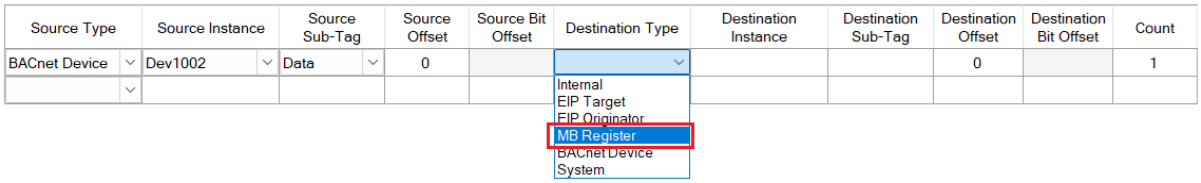


Figure 3.42 – IDS Copy - Modbus Destination Type

The ***Destination Instance*** will be the Modbus register type required.

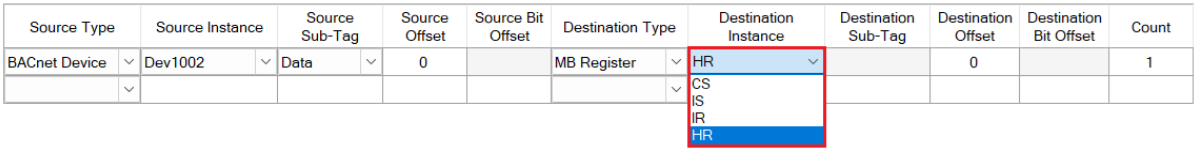


Figure 3.43 – IDS Copy - Modbus Destination Instance

The ***Destination Offset*** is the Modbus Register offset to where the data must be copied. The ***Count*** is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

3.7.2.2. MODBUS AUXILIARY MAP

The Modbus Auxiliary Map configuration is shown in the figure below. The Modbus configuration is only applicable when the module has a Modbus Client operating interface. Up to 100 mapping items can be configured. Within the 100 mapping items, a maximum of 20 Modbus TCP Server devices can be configured.

The Modbus Aux Map will be executed in a sequential manner, and a mapped item will be executed at the *Update Rate* in the Modbus parameters. That is, the *Update Rate* is the time between two successive mapped item executions.

The Modbus Auxiliary Map configuration window is opened by either double clicking on the module in the tree or by right-clicking the module and selecting ***Configuration***.

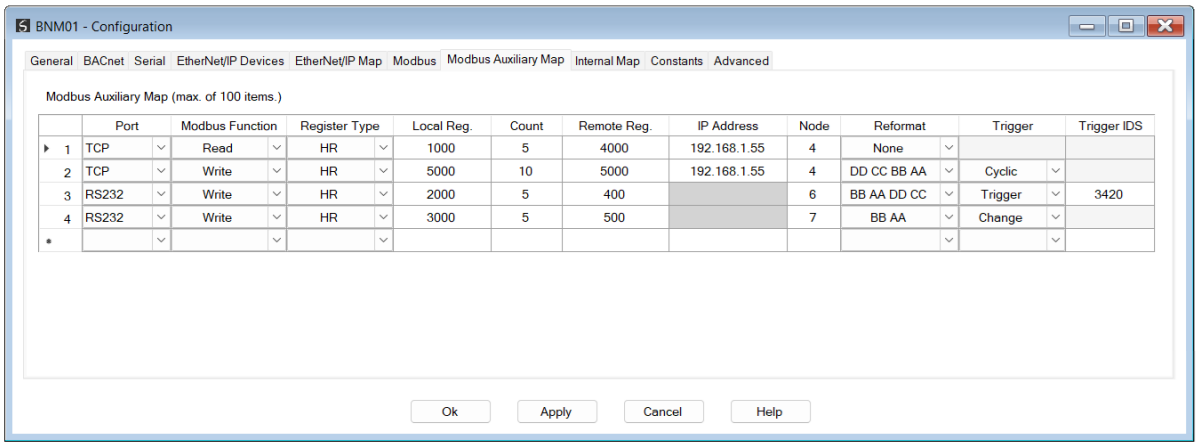


Figure 3.44 – Modbus Auxiliary Map Configuration

The Modbus Auxiliary Map configuration consists of the following parameters:

Parameter	Description
Port	The Modbus port to be used, either **TCP** – Modbus-TCP (Ethernet) **RS232** – Modbus RTU-232, **RS485** – Modbus RTU-485 **NOTE**: RS485 cannot be used if BACnet MS/TP has been enabled.
Modbus Function	This is the Modbus function that is sent to the Modbus Server.

	Read – Read a Modbus Register (e.g. HR, IR, CS, or IS) from a Modbus Server. **Write** – Write a Modbus Register (e.g. HR or CS) to a Modbus Server.
Register Type	Modbus Register Type: **CS** – Coil Status **IS** – Input Status **IR** – Input Register **HR** – Holding Register
Local Reg.	The local (internal) BACnet Master Modbus register address (0-4999).
Count	The number of Modbus elements to read or write.
Remote Reg.	The remote Server Modbus address register.
IP Address	The IP address of the remote Modbus Server.
Node	The Modbus Node address of the remote Modbus Server.
Reformat	Used to specify how the data is formatted before writing to, or after reading from, the Modbus Server. **None** – No reformatting applied. (AA BB CC DD). **BB AA** – 16bit Byte swap **BB AA DD CC** – 32bit Byte Pair Swap **CC DD AA BB** – Word Swap **DD CC BB AA** – Word and Byte Pair Swap
Trigger	For Modbus **Write** Functions, the user can select one of three triggers for the write operation. **Cyclic** The specific mapped item will be executed at the update rate (similar to all the read functions). **Change** When any of the data has been changed in the Modbus register range used to write the mapping, then this mapped item will be executed. **Trigger** When Trigger is selected, then the user will need to provide an IDS register which is used to trigger the specific Modbus Auxiliary option. When the provided IDS register changes, then the mapped item transaction will execute.
Trigger IDS	The IDS Register Offset used to trigger the respective Mapped item.

Table 3.4 – Modbus Auxiliary Map parameters

3.7.3. ETHERNET/IP ORIGINATOR

The BACnet Master can operate as an EtherNet/IP connection originator for cyclic (Class 1) or explicit (Class 3 or UCMM) data exchange. The explicit messaging can be configured in the *EtherNet/IP Devices* and *EtherNet/IP Map* in the Master configuration while the cyclic class 1 connections are added to the *EtherNet/IP Connections* node under the module in the Slate project tree.

3.7.3.1. ETHERNET/IP CLASS 1 DEVICE CONNECTIONS

The BACnet Master can establish up to 10 cyclic Class 1 EtherNet/IP connections to EtherNet/IP devices. This can be done by either manually entering the connection data into the Connection Parameter window, or by importing the configuration from one or more of the following sources:

- Online Logix Controller
- Logix Controller L5X
- EDS File
- Connection Library

A. MANUAL CONFIGURATION

A class 1 connection can be added to the *EtherNet/IP Connections* tree by right-clicking on the tree in Slate and selecting ***Add EtherNet/IP Connection***.

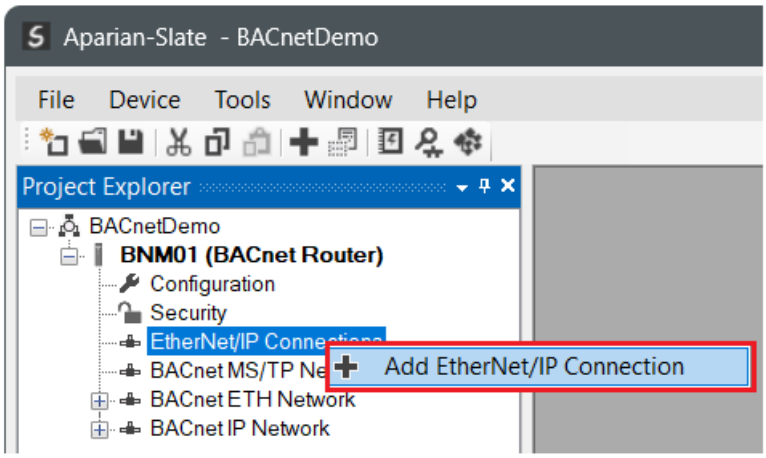


Figure 3.45 – Adding EtherNet/IP Class 1 Connection

Next the user will need to enter the connection parameters for the Class 1 connection.

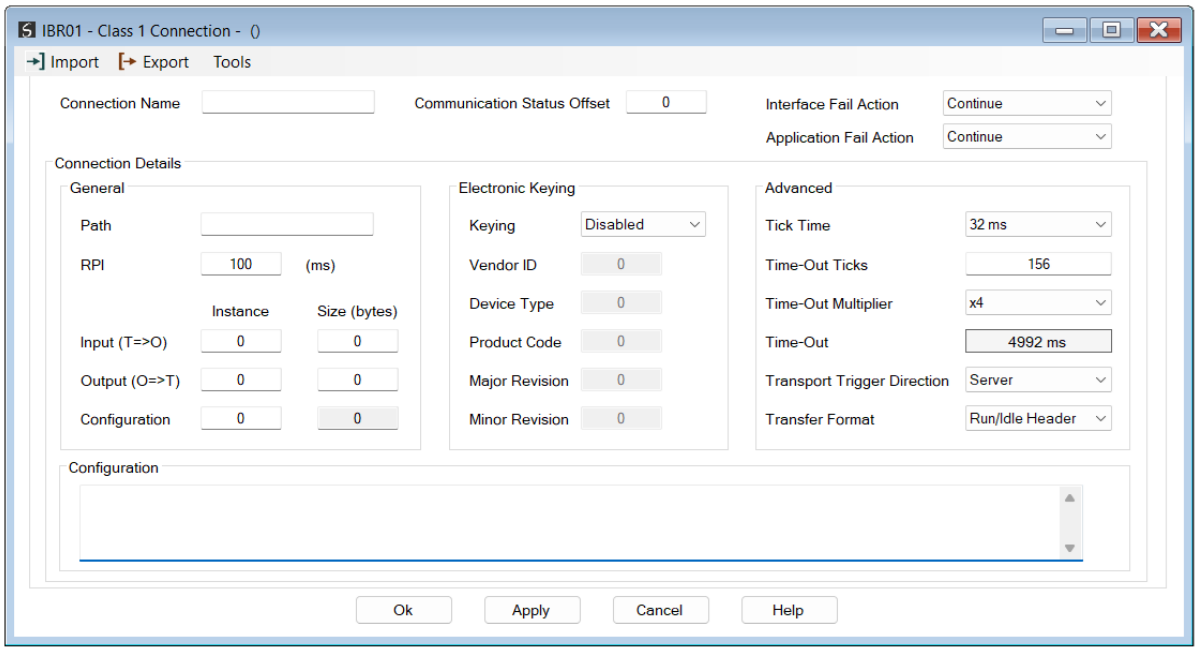


Figure 3.46 – EtherNet/IP Class 1 Connection Parameters

NOTE: It is recommended that the user not change the values in the *Advanced* frame of the connection parameters.

Parameter	Description
Connection Name	The instance name given to the Class 1 Connection.
Interface Fail Action	When the BACnet communication has failed, the EtherNet/IP IO can be configured to either keep the connection running as is, change the connection status to program mode, or force the connection offline. This will allow the EtherNet/IP device to go into a pre-determined state when the BACnet communication fails.
Application Fail Action	This field determines the behavior of the BACnet communication when this EtherNet/IP communication fails. **Continue** BACnet communication continues using existing data. **Zero Data** BACnet communication continues, but the mapped data for this EtherNet/IP connection is set to zero. **Offline** The BACnet communication is stopped.
General	
Path	The path to the target device. If the device is an Ethernet device, then this will just be the IP address of the module.

	If the device is, for example, a module in a backplane or via an adapter, then the user will need to enter the IP address of the bridge or adapter followed by the backplane port (for example 1) and the slot number of the device. Each item is separated by a comma. As an example, to connect to an Allen-Bradley Flex module (via the Flex Adapter at IP address 192.168.1.100) that is in slot 2 of the Flex backplane, the user will need to enter the following path: 192.168.1.100,1,2 (IP address, port (backplane), slot).
RPI	The requested packet interval (RPI) is the rate in milliseconds at which the data will be sent from the originator to the target and vice versa.
Input (T=>O) – Instance	The instance of the input assembly.
Input (T=>O) – Size (bytes)	The size in bytes of the input assembly.
Output (O=>T) – Instance	The instance of the output assembly.
Output (O=>T) – Size (bytes)	The size in bytes of the output assembly.
Configuration – Instance	The instance of the configuration assembly.
Configuration – Size (bytes)	The size in bytes of the configuration assembly. **NOTE:** This is a read-only value and will be equal to the number of bytes entered into the configuration window below.
Electronic Keying	
Keying	Electronic Keying can be used to ensure that the target device is the correct device type. **Disabled** Keying is not enabled, and no key information will be sent in the connection establishment. **Compatible** Keying has been enabled with compatibility enabled. This will allow devices with older firmware to also establish a connection. **Exact** Keying has been enabled and the exact device with specific firmware revision will allow the establishment of the connection.
Vendor ID	The Vendor ID of the target device.
Device Type	The Device Type of the target device.
Product Code	The Product Code of the target device.
Major Revision	The Major Revision of the target device.
Minor Revision	The Minor Revision of the target device.
Advanced (Note: Changing these values is not recommended)	
Tick Time	For unconnected messages, this is the time for each tick to calculate the unconnected Time-Out time.
Time-Out Ticks	The number of ticks before the unconnected message is set for timeout.

Time-Out Multiplier	This is the multiplier of the RPI to define the connection timeout time.
Time-Out	The unconnected message timeout time (read-only)
Transport Trigger Direction	The Transport Trigger direction; **Server** or **Client**.
Transfer Format	The Real Time Transfer Format for the O->T transfer. **Run/Idle Header** Includes a 16-bit sequence count, a 32-bit header with the Run/Idle notification, and the application data. **Modeless** Includes a 16-bit sequence count and the application data. **Heartbeat** Includes a 16-bit sequence count. **Zero-Length Idle** Includes a 16-bit sequence count and the application data only when not not in idle mode.
Configuration	
Data	The configuration data that is sent with the forward open connection establishment. The data will need to be entered as a space-delimited, hexadecimal string. For example: 0A 0D 12 EE The configuration size will increase by one each time a byte is added to the configuration.

Table 3.5 – EtherNet/IP Class 1 Connection Parameters

B. IMPORT FROM ONLINE CONTROLLER

Here the EtherNet/IP connection parameters are imported directly from an online Logix controller.

PREPARATION

Before the connection information can be imported, some preparation is required using Studio5000 and a Logix controller:

1. In Studio5000 create a new project and add the required EtherNet/IP device in the IO tree. If the device's profile supports configuration, then configure the device as required.
2. Download the project to a Logix controller.



NOTE: When instantiating modules in Studio5000 do not make use of the "Rack Optimization" communication format.

NOTE: Some versions Logix (V32+) do not support the reading of the module's configuration. Where possible use an earlier version (e.g. V24).

NOTE: It is possible that not all the connection information will be imported as it may not be available due to the type of device and Logix version.

IMPORT CONNECTION PARAMETERS

The connection parameters can be imported from the Logix controller by selecting the ***Import from Online Controller*** option located under the ***Import*** menu of the Class 1 Connection form.

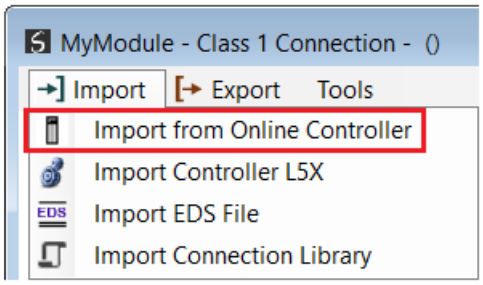


Figure 3.47 – Import from Online Controller

The Import Connection Parameters form will open.

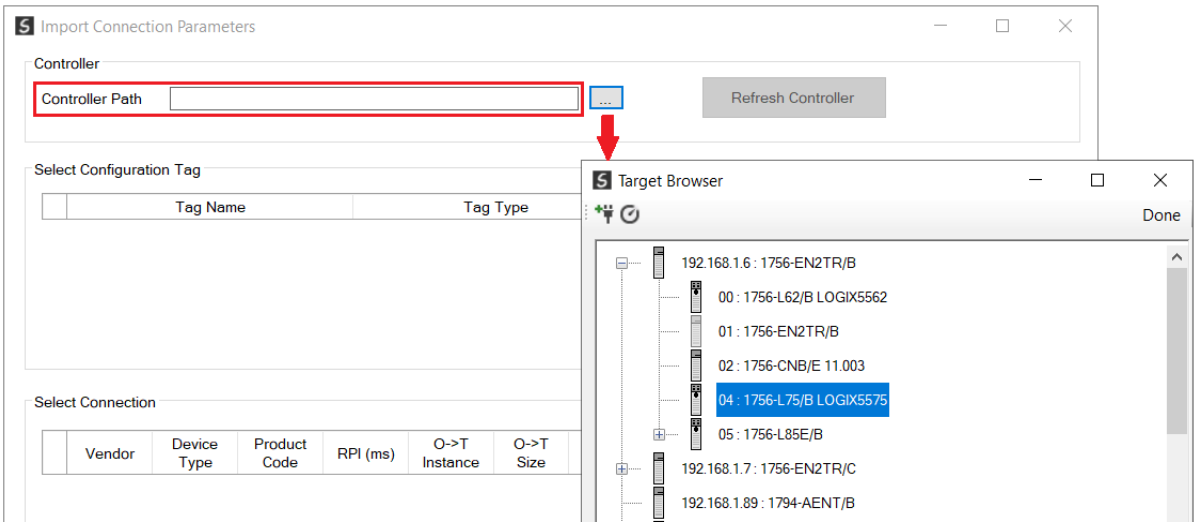


Figure 3.48 – Import Connection Parameters – Controller Path

Enter the path to the Logix controller. This can be either entered manually, or the Browse button "…", can be selected to launch the Target Browser, where the Logix controller can be selected.

Once the Logix controller path has been selected, all the device configuration tags and device connections will be read from the controller and displayed in the Configuration Tag grid and Connection grid respectively.

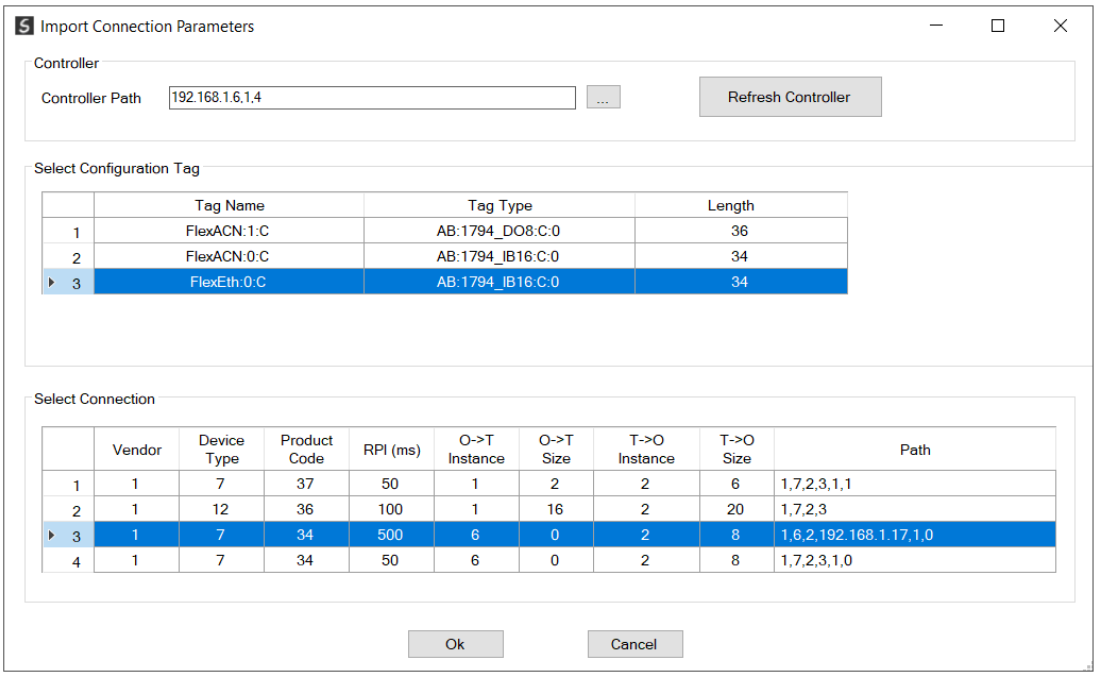


Figure 3.49 – Import Connection Parameters – Select Connection

In order to import all the necessary connection information, the user will need to select both the appropriate ***Configuration Tag***, and the matching ***Connection***.

The new connection's configuration data is derived from the selected ***Configuration Tag***, when the new connection's parameters are derived from the selected ***Connection***.

Once the appropriate selections have been made, press ***Ok***. The imported data will be populated into the Connection form.

The user can then modify the ***Connection Name***, ***Path*** and ***RPI*** as required.

C. IMPORT FROM CONTROLLER L5X FILE

Here the EtherNet/IP connection parameters are imported from a Logix controller's L5X file.

PREPARATION

Before the connection information can be imported some preparation is required using Studio5000:

1. In Studio5000 create a new project and add the required EtherNet/IP device in the IO tree. If the device's profile supports configuration, then configure the device as required.
2. Save the Studio5000 project as an L5X file.

NOTE: When instantiating modules in Studio5000 do not make use of the "Rack Optimization" communication format.

NOTE: It is possible that not all the connection information will be imported as it may not be available in the L5X file due to the type of device and Logix version.

IMPORT L5X FILE

The connection parameters can be imported from the L5X file by selecting the ***Import Controller L5X*** option located under the ***Import*** menu of the Class 1 Connection form.

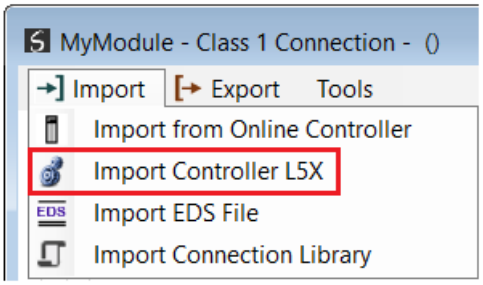


Figure 3.50 – Import from Controller L5X

The Import Connection Parameters form will open.

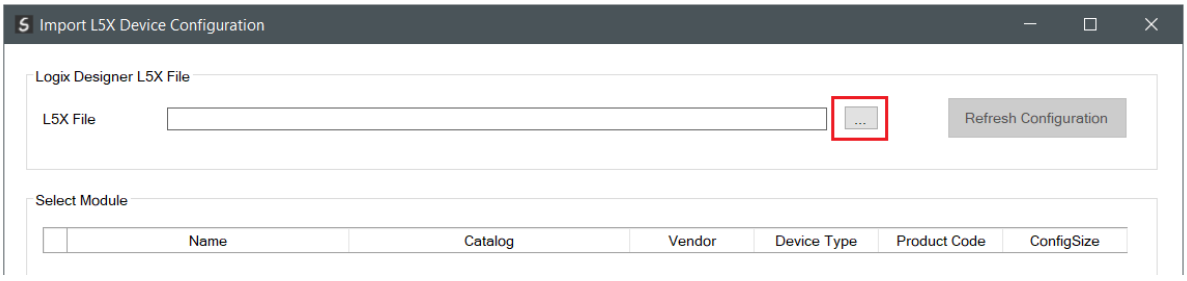


Figure 3.51 – Import L5X Device Configuration – Select L5X

Click on the Browse ("…") button to select the previously generated L5X file.

The modules found in the selected L5X file will then be displayed in the Module List.

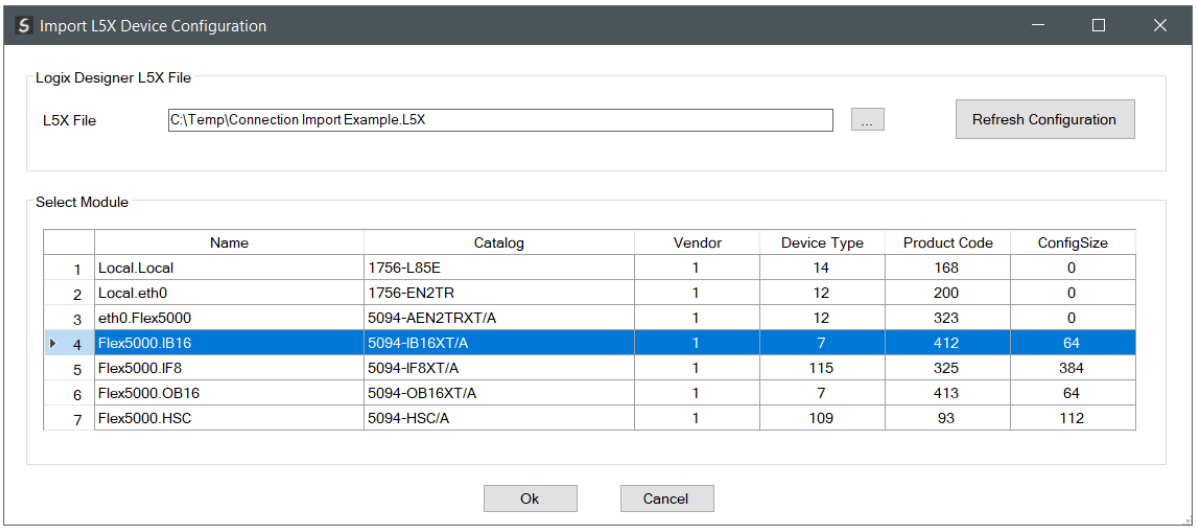


Figure 3.52 – Import L5X Device Configuration

Select the required module and click ***Ok***. The imported data will be populated into the Connection form.

The user can then modify the ***Connection Name***, ***Path*** and ***RPI*** as required.

D. IMPORT EDS FILE

The connection parameters can be imported from a suitable EDS file. Typically, this approach is preferred for devices that do not require configuration data.

To import the connection parameters from a device EDS file, select the ***Import EDS File*** option located under the ***Import*** menu of the Class 1 Connection form.

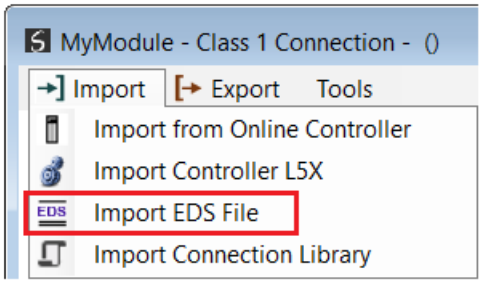


Figure 3.53 – Import EDS File

A File Open dialog will open allowing the user to select the EDS file.

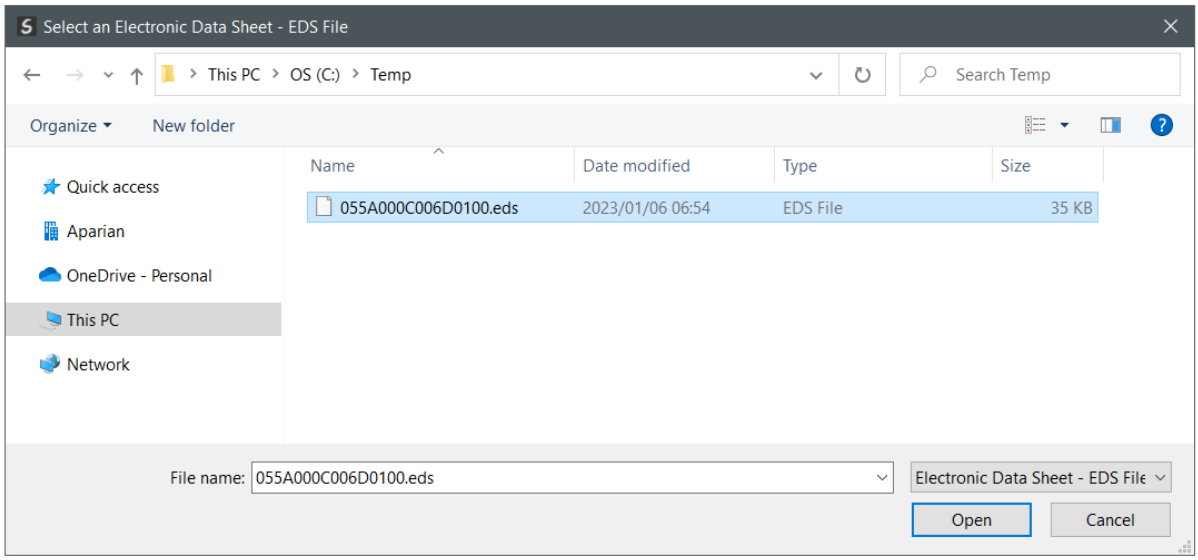


Figure 3.54 – Browse to EDS File

The selected EDS file will be imported, and a summary of the connections displayed. The user will need to select one of the IO connections.

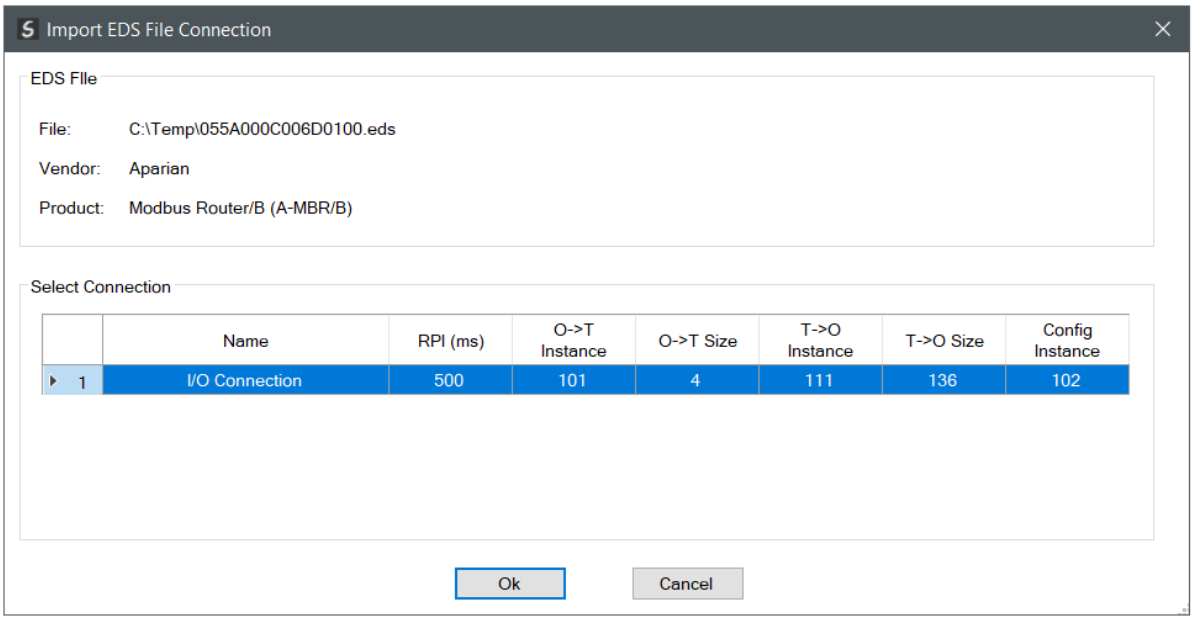


Figure 3.55 – Select Connection

The selected connection within the EDS file will be used to populate the Connection parameters.

The user can then modify the ***Connection Name***, ***Path*** and ***RPI*** as required.

E. IMPORT CONNECTION LIBRARY

The connection parameters can be imported from a previously created Connection Library (.EIPCNX) file.

NOTE: Please contact support to receive a pack of the latest Connection Library files, for commonly used devices.

To import the connection parameters from a Library file, select the ***Import Connection Library File*** option located under the ***Import*** menu of the Class 1 Connection form.

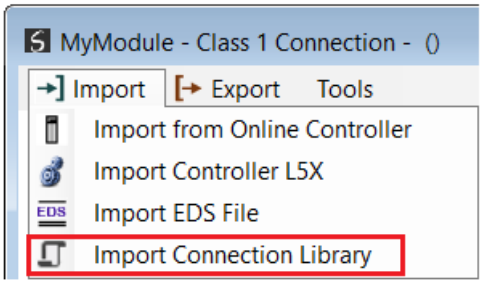


Figure 3.56 – Import Connection Library File

A File Open dialog will open allowing the user to select the Library (.EIPCNX) file.
The selected Library file will be used to populate the Connection parameters.

The user can then modify the ***Connection Name***, ***Path*** and ***RPI*** as required.

EXPORT LIBRARY FILE

In order to create a Library file for future use, select the ***Export Connection Library*** option located under the ***Export*** menu.

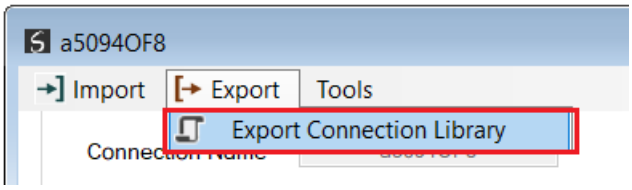


Figure 3.57 – Export Connection Library File

3.7.3.2. ETHERNET/IP EXPLICIT MESSAGE DEVICE CONNECTIONS

Up to 10 EtherNet/IP devices can be added for explicit messaging. The user will need to add each device as explained in the EtherNet/IP Devices section below. Once the EtherNet/IP

devices have been added the user can then configure the required mapping for the EtherNet/IP Explicit messaging as shown in EtherNet/IP Map section below.

A. ETHERNET/IP DEVICES

This tab is enabled when the Primary Interface is set to *EtherNet/IP Originator*.

The EtherNet/IP Devices configuration is shown in the figure below. Up to 10 EtherNet/IP devices can be configured with up to 50 EtherNet/IP mapped items allowing for either explicit EtherNet/IP Class 3 or Unconnected Messaging (UCMM) to any of the 10 configured devices. The data from each EtherNet/IP device is written to, or read from, an Internal Data Space with a size of 100Kbytes. See the *Explicit EtherNet/IP Messaging* section for more details.

The EtherNet/IP Devices configuration window is opened by either double clicking on the module in the tree, or by right-clicking the module and selecting *Configuration*.

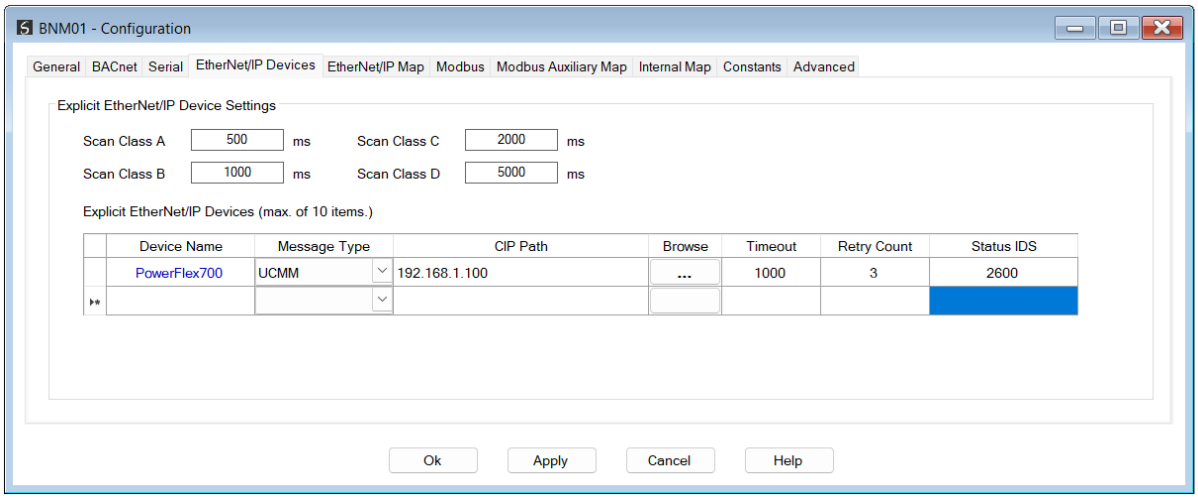


Figure 3.58 – EtherNet/IP Devices - Configuration

The EtherNet/IP Devices configuration consists of the following parameters:

Parameter	**Description**
Scan Class A, B, C, D	The configurable update rates (in milliseconds) for each scan class in the EtherNet/IP Map.
Device List (per device)	
Device Name	The user assigned instance name for the specific device.
Message Type	The module can use either ***Class 3*** or ***Unconnected Messaging*** when communicating to the target EtherNet/IP device.

CIP Path	The CIP Path to the target device. This can either be entered manually, or the user can browse to them by clicking the ***Browse*** button. The ***Target Browser*** will open and automatically scan for all available EtherNet/IP devices. If the EtherNet/IP module is a bridge module, it can be expanded by right-clicking on the module and selecting the ***Scan*** option. The required EtherNet/IP device can then be chosen by selecting it and clicking the ***Ok*** button, or by double-clicking on the target module.
Timeout	The amount of time (in milliseconds) the module will wait for a response from the target EtherNet/IP device.
Retry Count	The number of message retries before the target EtherNet/IP device is considered offline.
Comm Status Offset	This is the offset in the Internal Data Space (used to map EtherNet/IP device data) which provides the communication status of each EtherNet/IP device. The Communication Status is as shown below: Bit 0 - (1:Device Online , 0:Device Offline) Bit 1 to 7 – Reserved.

Table 3.6 – EtherNet/IP Devices configuration parameters

B. ETHERNET/IP MAP

This tab is enabled when the Primary Interface is set to *EtherNet/IP Originator*.

The EtherNet/IP Map configuration is shown in the figure below. Up to 50 EtherNet/IP mapped items, either explicit EtherNet/IP Class 3 or Unconnected Messaging (UCMM) to any of the 10 pre-configured devices can be configured. The data from each EtherNet/IP device is written to or read from Internal Data Space with a size of 100Kbytes. See the *Explicit EtherNet/IP Messaging* section for more details.

The EtherNet/IP Map configuration window is opened by either double clicking on the module in the tree, or by right-clicking the module and selecting *Configuration*.

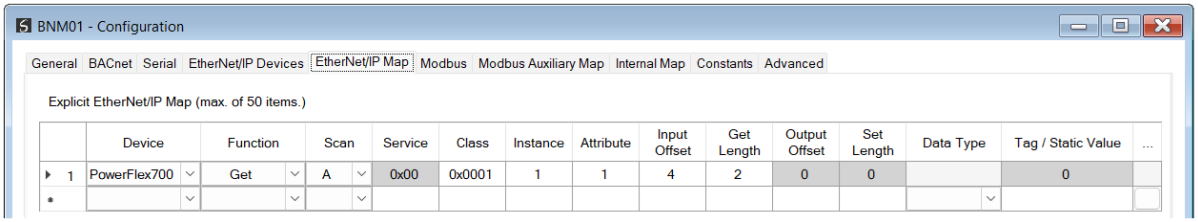


Figure 3.59 – EtherNet/IP Map configuration

The EtherNet/IP Map configuration consists of the following parameters:

Parameter	Description
Device	The device instance name configured in the previous EtherNet/IP Devices tab. The selected device will be used for executing the communication function.
Function	The user can select one of four functions. **Get** The module will read data from the target EtherNet/IP device by using the Get Single Attribute CIP function. The received data will be placed into the Internal Data Space at the ***Input Offset*** location configured in this tab. **Set** The module will write data to the target EtherNet/IP device by using the Set Single Attribute CIP function. The data to be written will be retrieved from the Internal Data Space at the ***Output Offset*** location configured in this tab. **Set Static** Similar to the Set function above, but the data to be written will be fixed (equal to the *Static Value*) parameter in this configuration window. This function will typically be used with the single (**S**) Scan class which means the BACnet Master module can be setup to write the fixed value only once when the target device communication has been established. **Custom** This function allows the user to use a custom Service to write and read data in the same transaction. The user will need to see which custom services that target device supports in that device's user manual. **Read Tag** When using a Logix controller as an EtherNet/IP Device, the BACnet Master module can read a Logix tag from the target Logix controller using Class 3 or UCMM messaging. The value from the tag will be saved at the configured Input Offset. **Write Tag** When using a Logix controller as an EtherNet/IP Device, the BACnet Master module can write to Logix tag from the target Logix controller using Class 3 or UCMM messaging. The value from the tag will be read from the configured Output Offset.
Scan	The user can select Scan Class **A**, **B**, **C** or **D** (which was configured in the EtherNet/IP Devices tab). The specific mapped item will then be executed at that configured scan class rate. The user can also select the **S** class which means that the mapped item will only execute once when communication to the target device is established. If the target device goes offline, then the mapped items with this class will be re-armed, and resent when communication is re-established.
Service	The custom CIP service/function which is only available when the ***Custom*** function has been selected.
Class, Instance, Attribute	The CIP class, instance, and attribute of the request message to be sent.
Input Offset	The location in the Internal Data Space where the received data will be written.

	This will only be available for ***Get*** and ***Custom*** functions.
Get Length	The length of the data to be received. If the number of bytes received is more than the Get Length, then the data will **not** be written to the Internal Data Space.  **NOTE:** When the function is Logix Read, then the Get Length will be the number of elements of the configured data type and **not** the byte count. This will only be available for ***Get*** and ***Custom*** functions.
Output Offset	The location in the Internal Data Space from where the data to be written to the target device will be read. This will only be available for ***Set*** and ***Custom*** functions.
Set Length	The length of the data to be written.  **NOTE:** When the function is Logix Write, then the Set Length will be the number of elements of the configured data type and **not** the byte count. This will only be available for ***Set*** and ***Custom*** functions.
Data Type	The data type of the Static Value. This will only be available for ***Set Static*** function.
Tag / Static Value	The value to be written to the target device when the ***Set Static*** function has been selected. **Note**: When using the SINT Array data type, the values must be entered as space-delimited hex values. For example: 05 34 2E A1

Table 3.7 – EtherNet/IP Map configuration parameters

3.7.3.3. INTERNAL DATA SPACE MAPPING

When the module is operating as an EtherNet/IP Originator, the data from the EtherNet/IP IO devices can be mapped to the BACnet interface using the Internal Map. The Internal Map configuration window is opened by either double clicking on the module in the tree or right-clicking the module and selecting *Configuration* and selecting the *Internal Map* tab.

A. IDS COPY – ETHERNET/IP ORIGINATOR SOURCE

When copying data from an EtherNet/IP IO to the BACnet interface, the ***Source Type*** needs to be ***EIP Originator***.

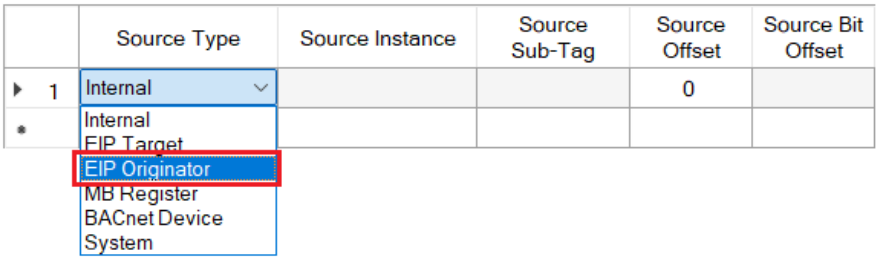


Figure 3.60 – IDS Copy – EtherNet/IP Originator Source Type

The ***Source Instance*** will be one of the EtherNet/IP IO devices previously added to the EtherNet/IP IO tree in Slate.

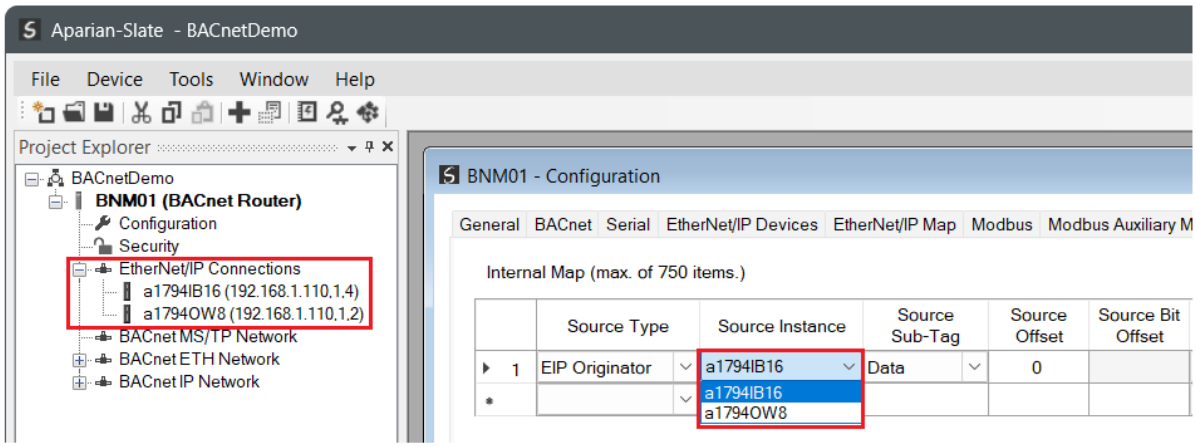


Figure 3.61 – IDS Copy – EtherNet/IP Originator Source Instance

The ***Source Offset*** is the offset in the selected EtherNet/IP device Class 1 **Input** Assembly. The ***Count*** is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation. The user can select either the Data, or Status, to copy from the EtherNet/IP connection.

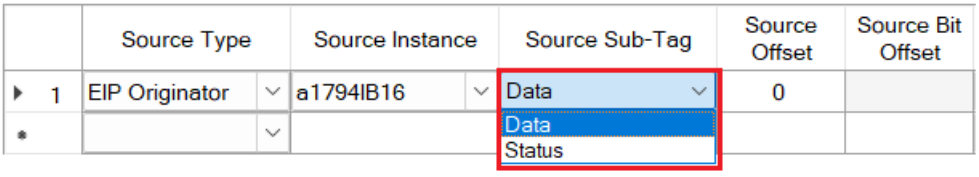


Figure 3.62 – IDS Copy – EtherNet/IP Originator Status

When selecting the Status, the format of the Status information is shown below:

Parameter	Data Type	Description
EtherNet/IP Originator Connection Status	DINT	Bit 0 – Connection Ok

Table 3.8 – EtherNet/IP Originator Connection Status

B. IDS COPY – ETHERNET/IP TARGET DESTINATION

When copying data from the BACnet interface to an EtherNet/IP IO device **Output** Assembly, the destination type needs to be EIP Originator.

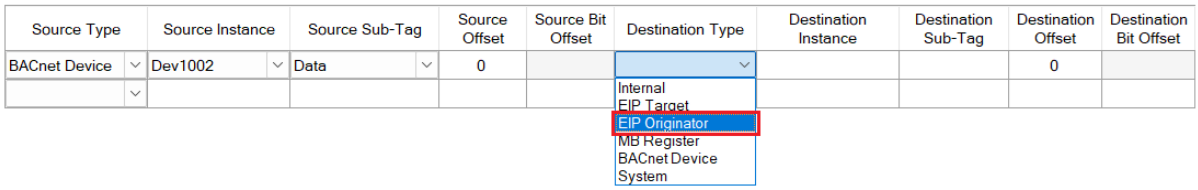


Figure 3.63 – IDS Copy – EtherNet/IP Originator Destination Type

The ***Destination Instance*** will be one of the EtherNet/IP IO devices added to the EtherNet/IP IO tree in Slate.

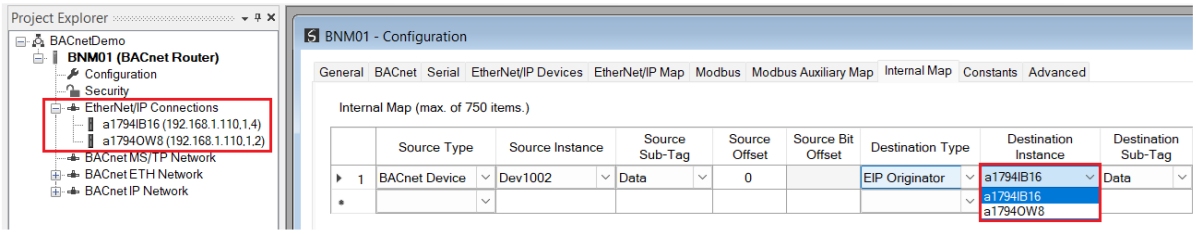


Figure 3.64 – IDS Copy – EtherNet/IP Originator Destination Instance

The ***Destination Offset*** is the offset in the selected EtherNet/IP device Class 1 **Output** Assembly. The ***Count*** is the number of **bytes** that will be copied. See the Internal Data Space Mapping section for more information regarding the operation.

3.8. BACNET DEVICES

BACnet devices can be added in two ways:

- Manual addition, where the device is configured manually and is usually done offline.
- Device Discovery, where the BACnet Master is online, and it automatically discovers devices on the enabled BACnet networks.

3.8.1. MANUAL ADDITION

To add a new BACnet device, right-click on the required BACnet network (MS/TP, BACnet/Ethernet or BACnet/IP) or Device Group, and select the ***Add Device*** option.

NOTE: If a BACnet Network is not enabled in the master configuration then it will not be visible in the project tree.

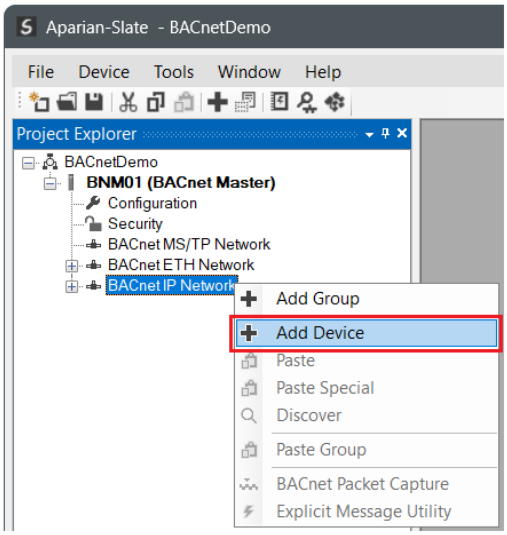


Figure 3.65 – Add BACnet device

The BACnet Device configuration form will open.

3.8.1.1. BACNET DEVICE CONFIGURATION – GENERAL

The BACnet Device - General configuration is shown in the figure below. The Device General configuration window is opened by either double clicking on the BACnet device in the project tree or right-clicking the BACnet device and selecting the ***Configuration*** option.

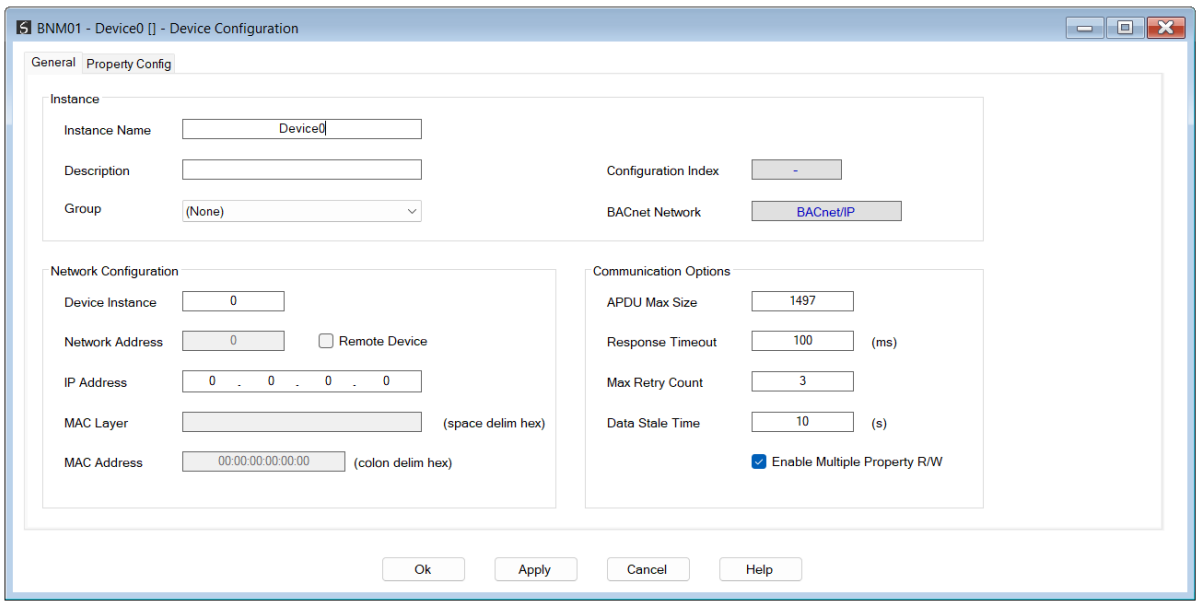


Figure 3.66 – New BACnet device configuration

The **BACnet device - General** configuration consists of the following parameters:

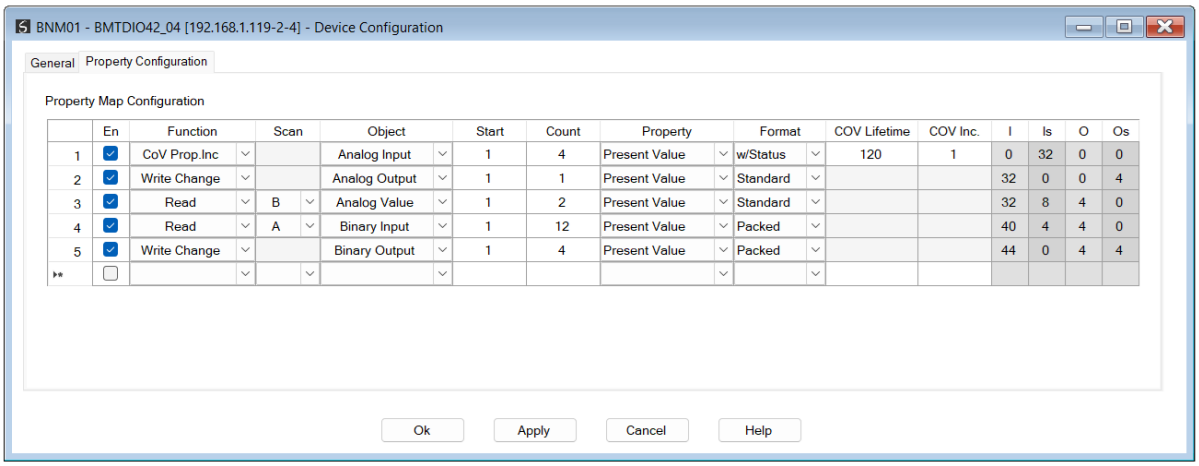
Parameter	Description
Instance	
Instance Name	User defined instance name for the device. This name is limited to 16 characters and must be unique across all BACnet networks. This name will be used for the creation of Logix tags when configured EtherNet/IP target.
Description	User defined description for the device. The description is limited to 32 characters.
Group	Optional device Group to which the device is assigned. The Group is used only to make the project tree more manageable and has no operational function.
Configuration Index	The Index (0-249) where this device is located in the master configuration. This parameter is read-only.
BACnet Network	The BACnet network type for this device. This parameter is set during instantiation and cannot be modified.
Network Configuration	
Device Instance	The unique 32-bit device instance number.
Network Address	The network address or node number. This is relevant only for **MSTP** devices, or devices bridging to MSTP networks.
Remote Device	This option must be selected only if the device is accessible via a BACnet bridge device.
IP Address	The IP address of the device. This is relevant only for **BACnet/IP** devices.
MAC Layer	The MAC layer contains the routing information to access a remote device via a bridge. This is relevant only to remote devices, and is entered as a space delimited HEX byte array.
MAC Address	The Ethernet MAC address of the device. This is relevant only for **BACnet/Ethernet** devices.
Communication Options	
APDU Max. Size	The maximum APDU size (bytes) that can be received by the device.
Response Timeout	The time (milliseconds) the master will wait for the device to reply, before attempting a retry.
Max. Retry Count	The maximum number of master request re-transmissions to this device before the transaction is deemed to have failed.
Data Stale Time	The time (seconds) elapsed since the last valid update of a device's property before it is deemed to be stale.

Enable Multiple Property R/W	Selects whether this device supports Multiple Property Read / Write requests. Enabling this feature can dramatically speed up property read and writes, however not all BACnet devices support this. **NOTE:** If the device is not able to accept the a multiple r/w operation, the master will then revert to individual read and write operations.

Table 3.9 - BACnet device – General configuration parameters

3.8.1.2. BACNET DEVICE – PROPERTY CONFIGURATION

The BACnet Device – Property configuration is shown in the figure below and is used to configure the how the required Object Properties are exchanged and mapped.



	En	Function	Scan	Object	Start	Count	Property	Format	COV Lifetime	COV Inc.	I	Is	O	Os
1	☑	CoV Prop.Inc		Analog Input	1	4	Present Value	w/Status	120	1	0	32	0	0
2	☑	Write Change		Analog Output	1	1	Present Value	Standard			32	0	0	4
3	☑	Read	B	Analog Value	1	2	Present Value	Standard			32	8	4	0
4	☑	Read	A	Binary Input	1	12	Present Value	Packed			40	4	4	0
5	☑	Write Change		Binary Output	1	4	Present Value	Packed			44	0	4	4
▸*	☐													

Figure 3.67 – New BACnet device Property configuration

The **BACnet device - Property** configuration consists of the following parameters:

Parameter	Description
En	Enable. When unchecked this item will be ignored by the BACnet master.
Function	The BACnet master operational function to be applied to the Object-Property combination: **Read** The object property is explicitly read at the configured scan rate. **Write** The object property is explicitly written at the configured scan rate. **Write Change** The object property is explicitly written only when the value changes. **CoV** The BACnet master registers a CoV (Change of Value) subscription with the device lasting the configured CoV Period. The subsequent unsolicited device reports for this object-property will be processed. The BACnet master will renew the CoV subscription each time the CoV Period elapses.

	CoV Prop. The BACnet master registers a CoV (Change of Value) subscription with the device for this specific property and lasting the configured CoV Period. The subsequent unsolicited device reports for this object-property will be processed. The BACnet master will renew the CoV subscription each time the CoV Period elapses. **CoV Prop. Inc** The BACnet master registers a CoV (Change of Value) subscription with the device for this specific property and lasting the configured CoV Period. The CoV increment value specifies the minimum change required in the Present Value before the device will report the value. The subsequent unsolicited device reports for this object-property will be processed. The BACnet master will renew the CoV subscription each time the CoV Period elapses. **Map** No explicit action is taken by the BACnet master. When this object-property is received through some other mechanism (e.g. Group Read), then the value will be processed and mapped accordingly.
Scan	The scan class (A-F) to be used for the explicit action. The scan classes are configured in the main BACnet master configuration. The scan class is relevant only for the Read and Write function.
Object	The specified BACnet object to which the property belongs, either: • Analog Input • Analog Output • Analog Value • Binary Input • Binary Output • Binary Value • Device • Group • Life Safety Point • Multi-State Input • Multi-State Output • Multi-State Value
Start	The starting Object Instance within the device.
Count	The number of contiguous Object Instances from the Start Index. This count is limited to 1-32. This option allows multiple Object-Property combinations with the same requirements, to be easily configured
Property	The Property of the configured Object to be exchanged. The choice of properties depends on the Object and Function selected. In most cases the ***Present Value*** is used.
Format	The format used to map the data to the primary interface.

	Standard Only the property data is mapped. **w/Status** The property data, and the associated property status is mapped to the primary interface. The property status includes flags indicating the transaction Success, Failure, Staleness as well as a 16-bit Success count. **Packed** This format is available only for Binary objects, and packs the property data into a single 32 bit word.
CoV Lifetime	Configures the time (seconds) that the Change of Value subscription will remain active before being renewed.
CoV Inc.	The CoV increment value (deadband) specifies the minimum change required in the Present Value before the device will report the (CoV) value.
I (Input Offset)	The data offset in the device's input data assembly associated with this object-property. This parameter is read-only.
Is (Input Size)	The data size (bytes) in the device's input data assembly associated with this object-property. This parameter is read-only.
O (Output Offset)	The data offset in the device's output data assembly associated with this object-property. This parameter is read-only.
Os (Output Size)	The data size (bytes) in the device's output data assembly associated with this object-property. This parameter is read-only.

Table 3.10 - BACnet device – Property configuration parameters

NOTE: A maximum of 256 object-property instances can be configured per BACnet device.

3.8.2. DEVICE DISCOVERY

BACnet devices can also be added online using the Device Discovery utility. This provides a fast method to not only instantiate devices, but also to auto-configure the device's object properties.

To open the Discovery utility, right-click on the required BACnet network and select the ***Discover*** option.

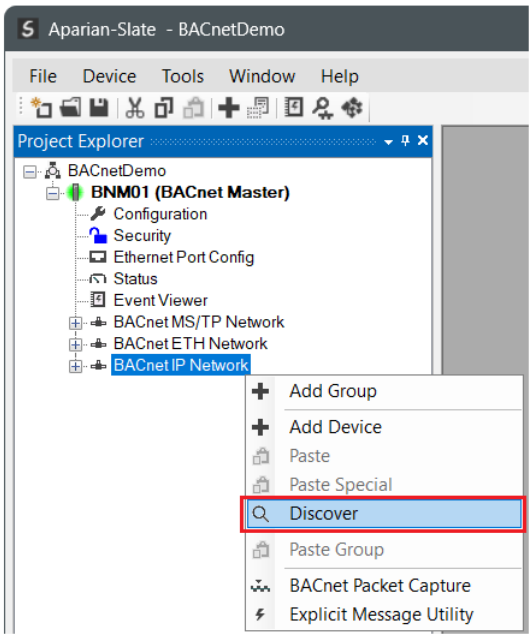


Figure 3.68 – Discover

The BACnet device ***Discover*** utility will open.

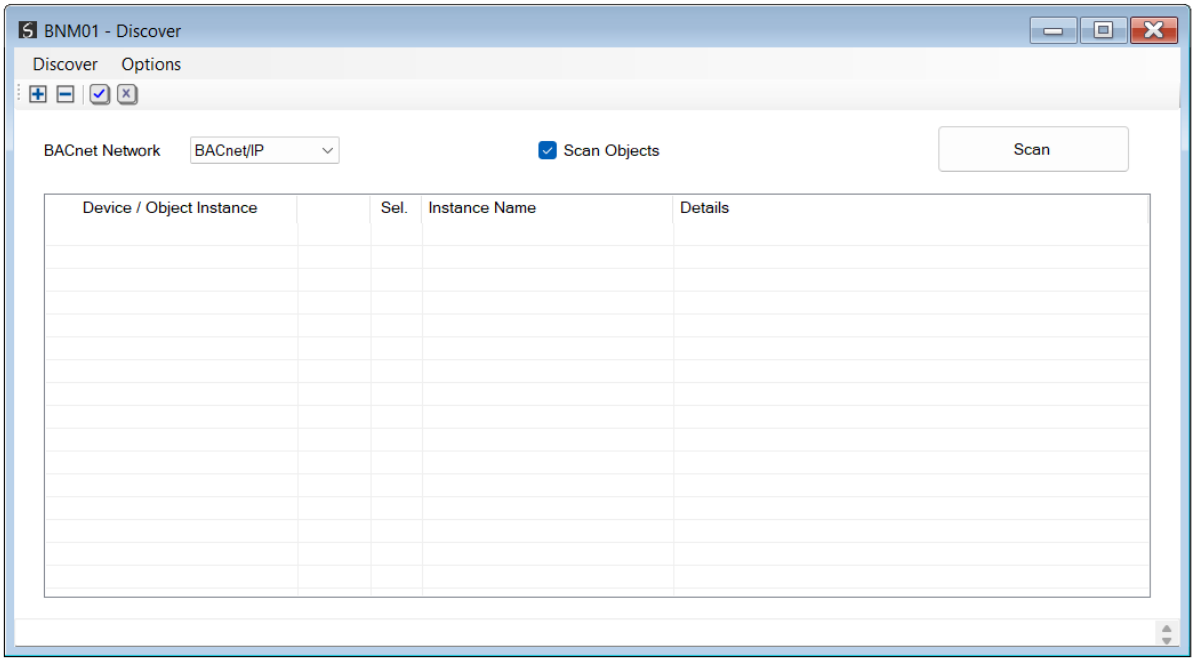


Figure 3.69 – Discover utility

The BACnet network will default to the selected network. This can be changed, but note that any previously discovered nodes will be cleared if the network changes.

3.8.2.1. DISCOVERY INSTANTIATION OPTIONS

Before initiating the scan, the ***Discovery Instantiation Options*** can be modified to adjust how the devices and their properties are automatically instantiated.

To configure these options select the ***Discovery Instantiation Options*** item under the ***Options*** menu.

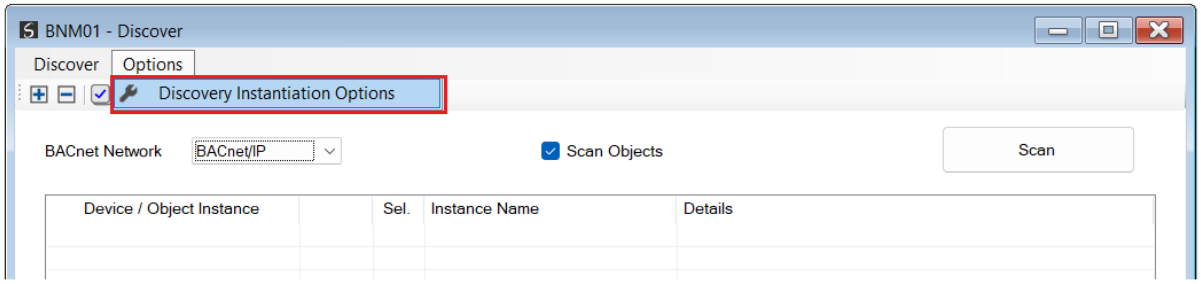


Figure 3.70 – Open Discovery Instantiation Options

The ***Discovery Instantiation Options*** form will open. The first tab (Property Mapping) allows the user to configure how each discovered object will be added to the device's property mapping. For each object type, the user can select the default Function, Scan class, Property and Format. In the case of CoV functions, the CoV Period and Increment can be configured.

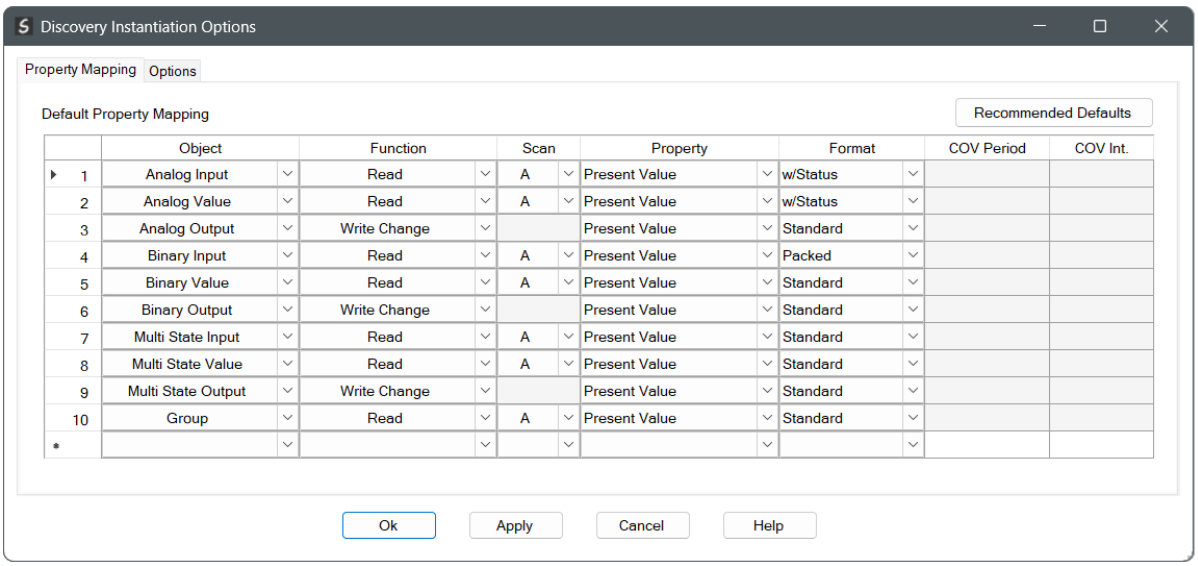


Figure 3.71 – Discovery Instantiation Options – Property Mapping

The second tab (Options) allows the user to specify the default communication settings for the instantiated devices.

These parameters include the default APDU Max. Size, Response Timeout, Max Retry Count, Data Stale Time and the Multiple Property R/W enable.

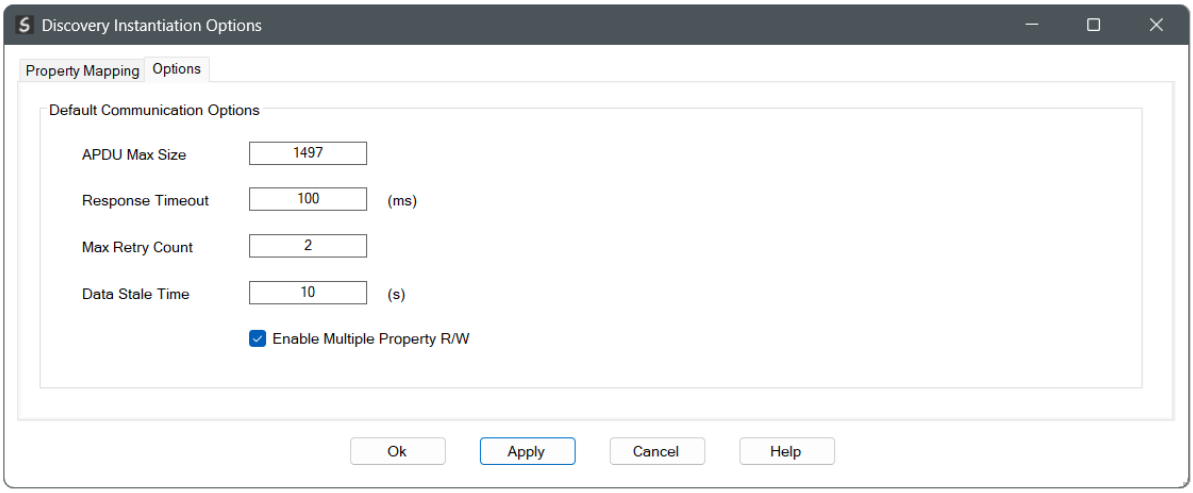


Figure 3.72 – Discovery Instantiation Options – Communication Options

NOTE: The Discovery Instantiation Options are saved in the offline Slate project for future use.

3.8.2.2. SCAN

If the ***Scan Objects*** checkbox is set, then each discovered device will be interrogated for the object instances that it supports. This provides valuable information, but increases the time required to complete the scan.

To initiate the scan, press the ***Scan*** button, or the ***Scan Network*** menu item, under the Discover menu.

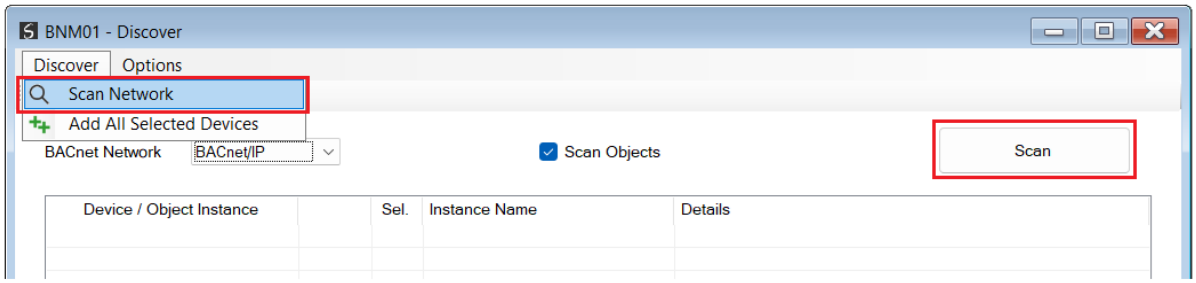


Figure 3.73 – Initiate Scan

The BACnet master will then scan the selected BACnet network for devices and display them.

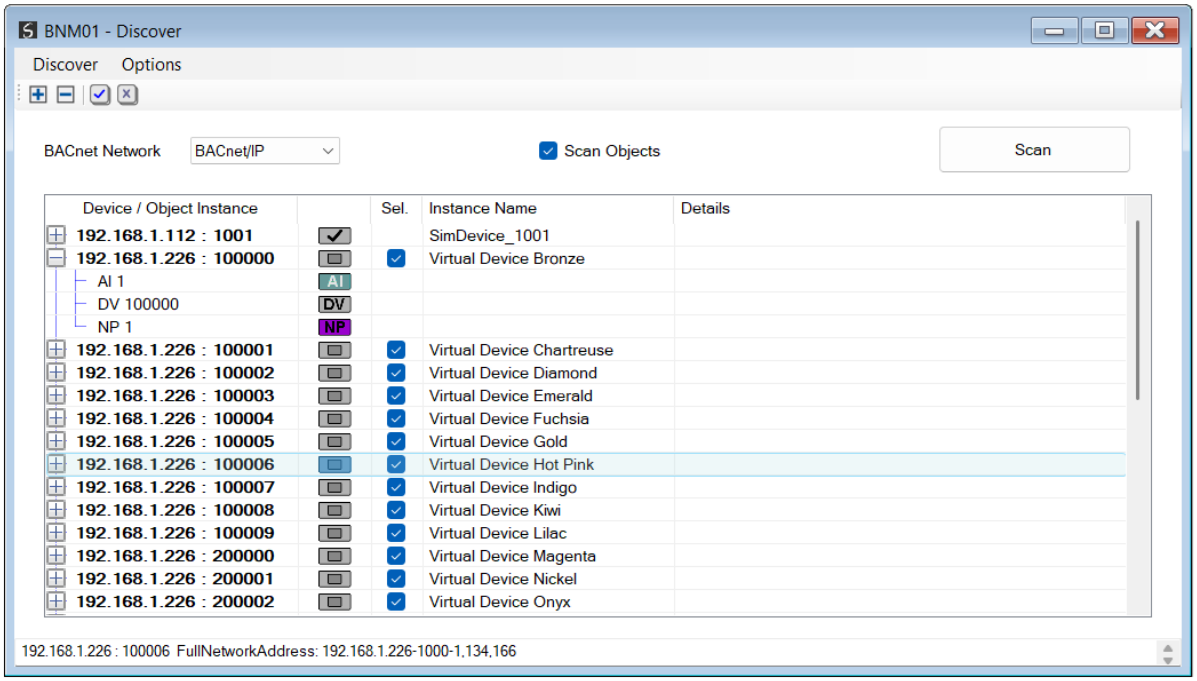


Figure 3.74 – Scan Results

Each device can be expanded to show its discovered objects. The following icons indicate the device / object types.

Icon	Description
	BACnet Device (Not in the master configuration)
	BACnet Device (Already added to master configuration)
	Device Object Instance
	Network Port Object Instance
	Analog Input Object Instance
	Analog Output Object Instance
	Analog Value Object Instance
	Binary Input Object Instance
	Binary Output Object Instance
	Binary Value Object Instance
	Multi-State Input Object Instance
	Multi-State Output Object Instance
	Multi-State Value Object Instance
	Group Object Instance

Table 3.11 – Scan Device/Object Icons

3.8.2.3. DISCOVERY DEVICE FUNCTIONS

A number of functions are available in the Discover tree. Right-click on a device or object will open the context menu.

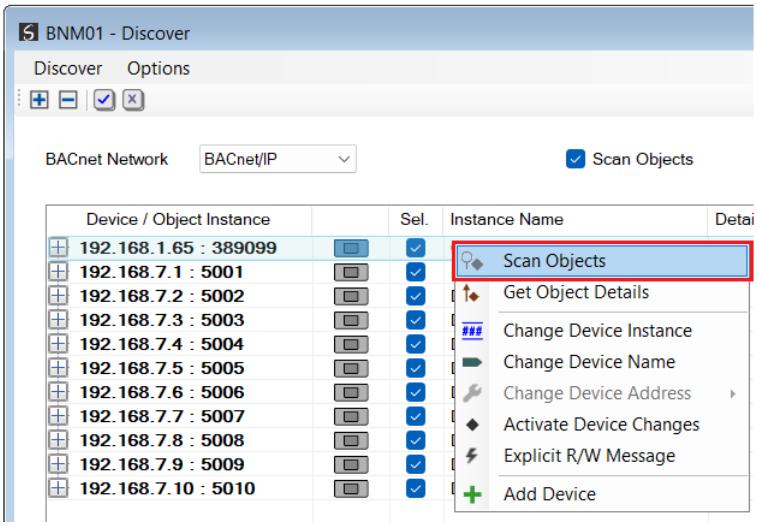


Figure 3.75 – Device / Object Functions

The following functions are available depending on the type of device or object selected:

A. SCAN OBJECTS

This option scans (or rescans) the selected device for all the objects it contains. This is not usually required unless the initial Scan did not include the ***Scan Objects*** option.

B. GET OBJECT DETAILS

This option reads basic information for the selected object. If a device is selected, then all the objects in the device will be read. The information includes instance name, and if applicable the Present Value and Engineering Units.

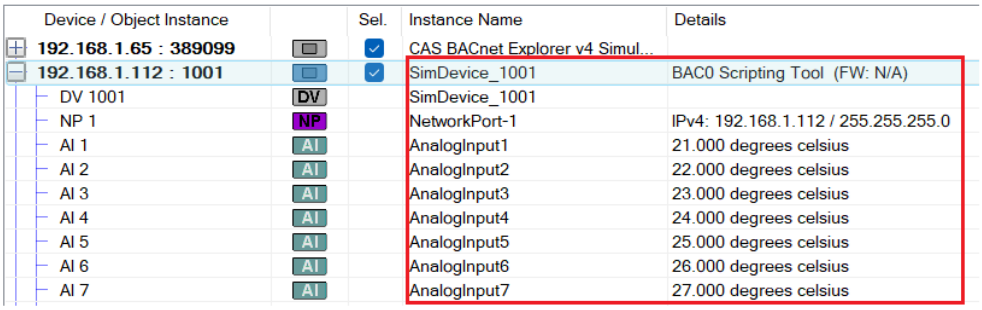


Figure 3.76 – Get Object Details

C. CHANGE DEVICE INSTANCE

The Device Instance can be changed by selecting the ***Change Device Instance*** menu option. The new ***Device Instance*** can then be entered.

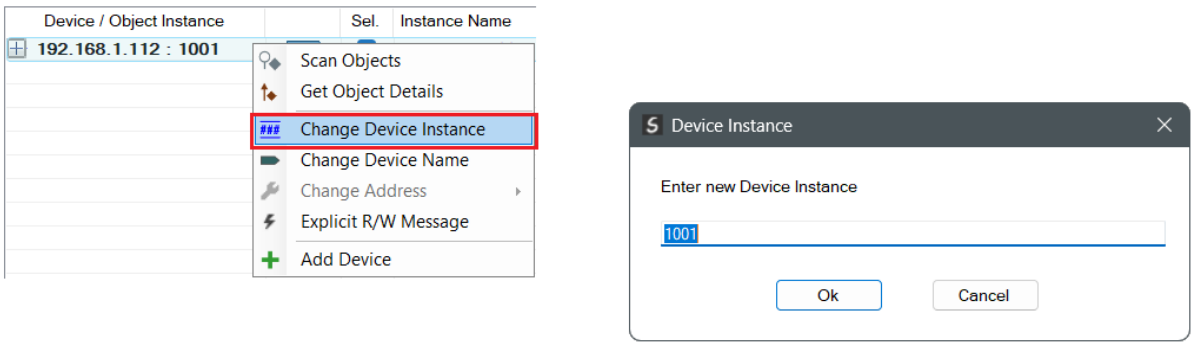


Figure 3.77 – Change Device Instance

NOTE: When entering a new Device Instance make sure that this value is unique for this network.

NOTE: Not all BACnet devices support this function and may require other vendor specific tools to adjust this value.

D. CHANGE DEVICE NAME

The Device Name can be changed by selecting the ***Change Device Name*** menu option. The new ***Device Name*** can then be entered.

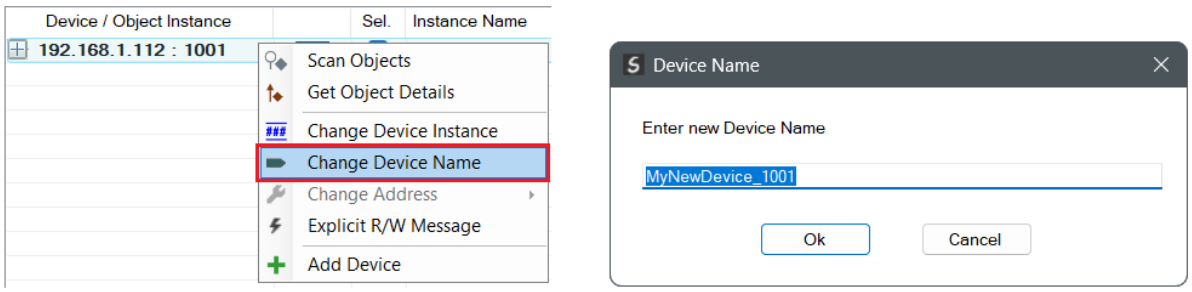


Figure 3.78 – Change Device Instance

NOTE: Not all BACnet devices support this function and may require other vendor specific tools to adjust this value.

E. CHANGE ADDRESS

Certain Network Address properties can be changed by right-clicking on the ***Network Port*** object under a device and selecting the ***Change Address*** option.

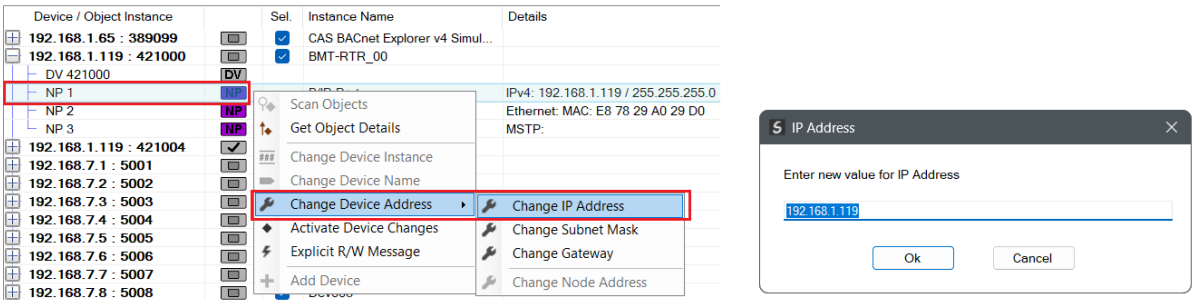


Figure 3.79 – Change Network Port – IP Address

In the case of a BACnet/IP device, this can be used to change the IP Address, Subnet Mask and Default Gateway address. In the case of a BACnet MS/TP device, this can be used to change the Node Address.

NOTE: Most devices will require the ***Activate Device Changes*** command to be sent before the IP address changes will take effect.

NOTE: Not all BACnet devices support this function and may require other vendor specific tools to adjust this value.

F. ACTIVATE DEVICE CHANGES

Following specific device changes (e.g. IP address change) an ***Activate Device Change*** command will need to be sent to the device for the changes to take effect. This can be actioned by right-clicking on the device and selecting the ***Activate Device Change*** menu option.

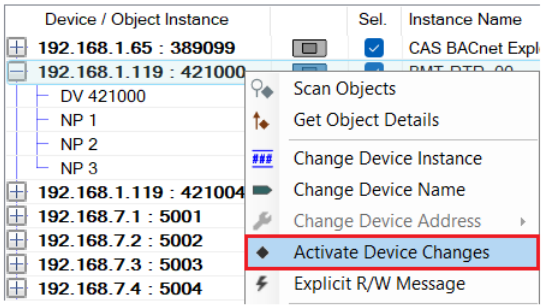


Figure 3.80 – Activate Device Changes

G. EXPLICIT R/W MESSAGE

The Explicit Read/Write utility can be used to read and write various parameters to the device. To select this utility, right-click on a device or object and select the ***Explicit R/W Message*** option.

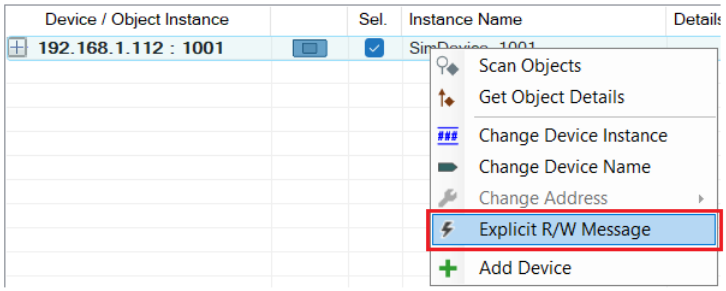


Figure 3.81 – Explicit R/W Message Utility

NOTE: See more details on the Explicit Message Utility in chapter 8.

H. ADD DEVICE - SINGLE

To add a single device, right-click on the device and select the Add Device option.

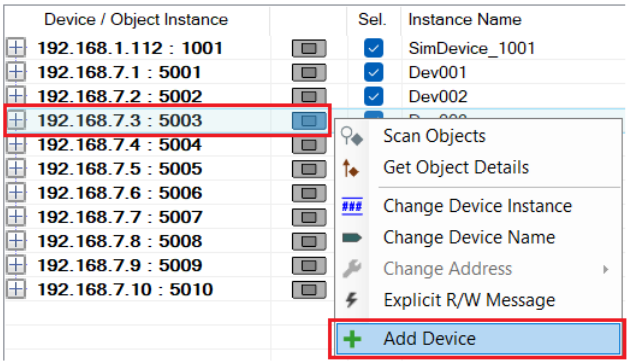


Figure 3.82 – Add Single Device

The BACnet device configuration form will be opened, with all the fields pre-populated with the parameters from the Device Discovery and in accordance with the Discovery Instantiation Options.

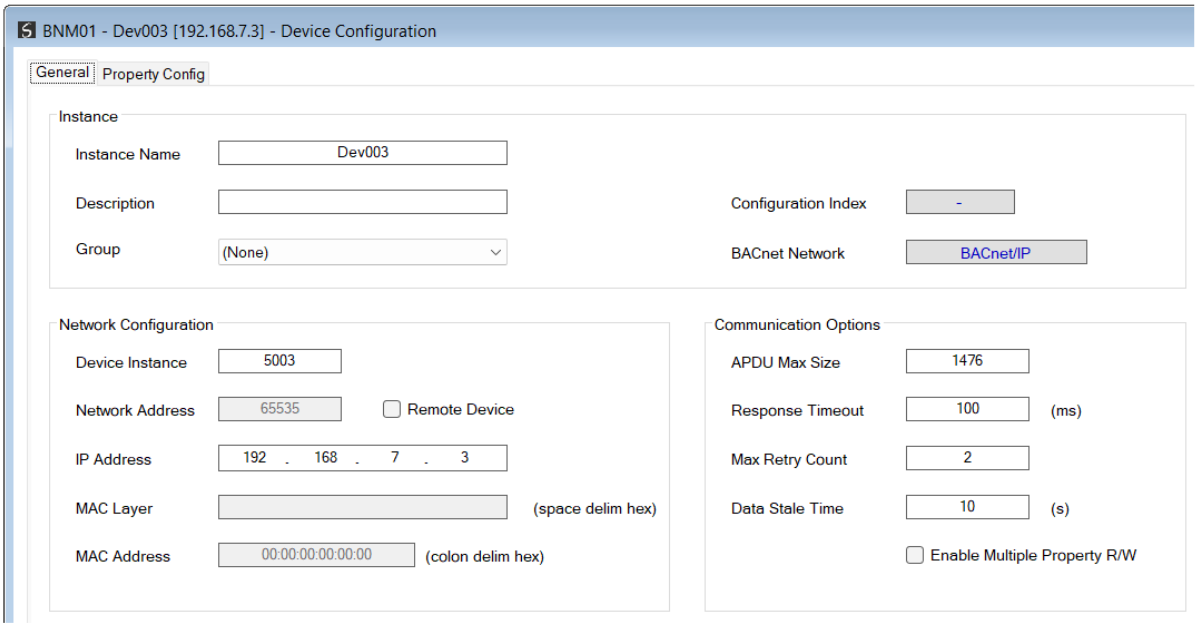


Figure 3.83 – Device Configuration

The device will only be added to the BACnet Master once it has been accepted by clicking the ***Apply*** or ***Ok*** button.

A. ADD DEVICE - MULTIPLE

Multiple devices can be added by first selecting the required devices (in the "Sel." Column) and then selecting the ***Add All Selected Devices*** menu item in the ***Discover*** menu.

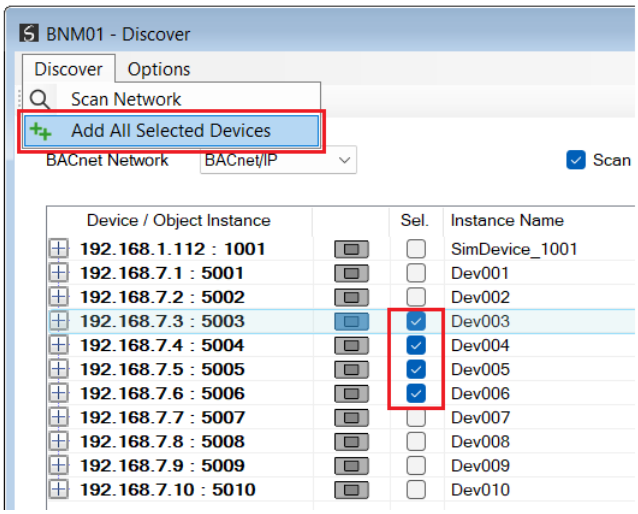


Figure 3.84 – Add All Selected Devices

NOTE: When adding multiple devices, all the selected BACnet devices will be immediately added to the BACnet Master configuration.

The configuration of each BACnet device will be a combination of the discovered parameters and objects from the Device Discovery and the Discovery Instantiation Options.

The following tool buttons can be used to assist with the selection and navigation process:

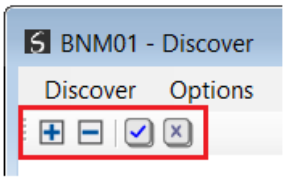


Figure 3.85 – Discover Tools

+	Expand all tree items
-	Collapse all tree items
✓	Select All devices
☒	Deselect All devices

3.9. INTERNAL DATA SPACE MAP

The internal data map is used to exchange data from the Ethernet interface to the BACnet interface and vice versa. Up to 750 items can be mapped. The Internal Map configuration window is opened by either double clicking on the module in the tree, or right-clicking the module and selecting ***Configuration*** and then selecting the ***Internal Map*** tab.

The ***Count*** is the number of bytes that will be copied from the source to the destination. There are four different ***Copy Functions*** that can be used.

Function	Description
Byte to Byte	Each **byte** from the source will be directly copied to each **byte** in the destination.
Byte to Bit	Each byte from the source will be copied to each bit in the destination. If a value greater than zero is read from the **source byte** then a 1 will be written to the **destination bit** address. If a value of zero is read from the **source byte** then a 0 will be written to the **destination bit** address. The destination offset will be the bit offset and the destination address will be incremented by one bit each time.
Bit to Bit	Each **bit** from the source will be directly copied to each **bit** in the destination.
Bit to Byte	Each bit from the source will be copied to each byte in the destination. If a value of one is read from the **source bit** then a 1 will be written to the **destination byte** address. If a value of zero is read from the **source bit** then a 0 will be written to the **destination byte** address. The source offset will be the bit offset and the source address will be incremented by one bit each time.

Table 3.12 – Internal Map Copy functions

The data in the destination source can also be reformatted. The reformat option provides five different reformat options.

NOTE: The reformat option is only available for ***Byte to Byte*** Copy Functions.

Function	Description
None	No reformatting applied (AA BB CC DD)
BB AA	16-bit Byte swap
BB AA DD CC	32-bit Byte Pair Swap
CC DD AA BB	Word Swap
DD CC BB AA	Word and Byte Pair Swap

Table 3.13 – Internal Map Reformat Options

3.9.1. COPY FROM

One of six sources can be selected to copy from:
Internal, EIP Target, EIP Originator, MB Register, BACnet Device, and System.

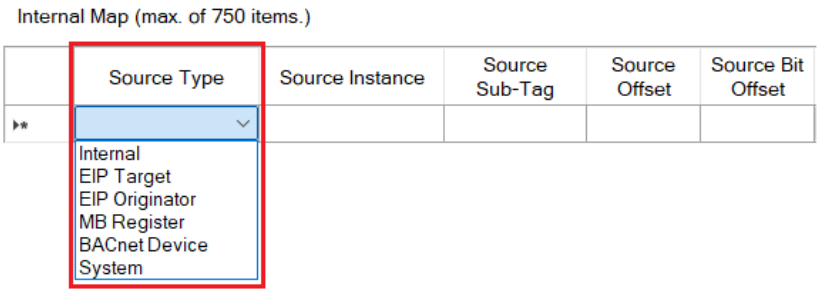


Figure 3.86 – Internal Map – Source Type

3.9.1.1. INTERNAL

When copying data from the internal data space (IDS), the source type needs to be Internal.

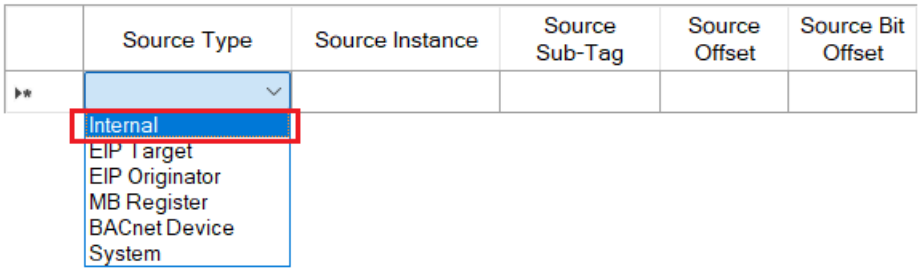


Figure 3.87 – IDS Copy – Internal Source Type

The Source Instance is Not Applicable for the internal data space. The ***Source Offset*** is the offset in the *Internal Data Space (IDS)* which has a maximum of 100 Kbytes. The ***Count*** is the number of **bytes** that will be copied.

3.9.1.2. EIP TARGET

When copying data from a connection originator (e.g. the output assembly from the Logix Controller) to the BACnet interface, the source type needs to be EIP Target.

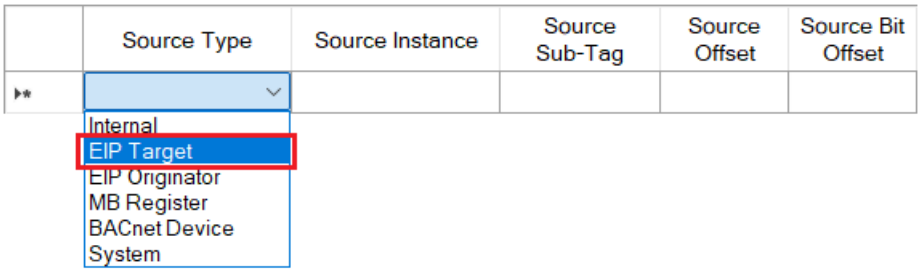


Figure 3.88 – IDS Copy – EtherNet/IP Target Source Type

The source instance will be the connection number, which can be connection 1 to 25, based on the number of connections configured. The Source Offset is the offset in the EtherNet/IP output assembly from where the data must be copied. The Count is the number of **bytes** that will be copied.

3.9.1.3. EIP ORIGINATOR

When copying data from an EtherNet/IP IO to the BACnet interface, the source type needs to be EIP Originator.

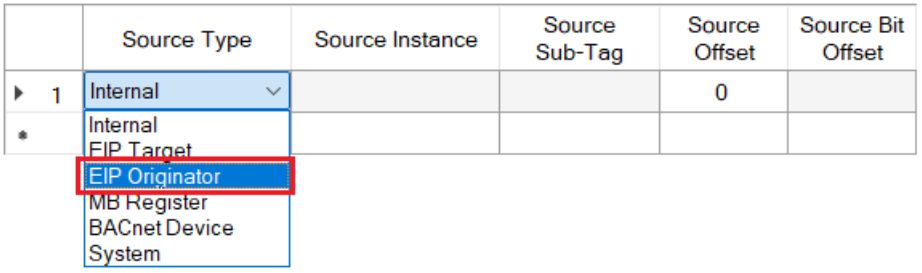


Figure 3.89 – IDS Copy – EtherNet/IP Originator Source Type

The source instance will be one of the EtherNet/IP IO devices added to the EtherNet/IP IO tree in Slate.

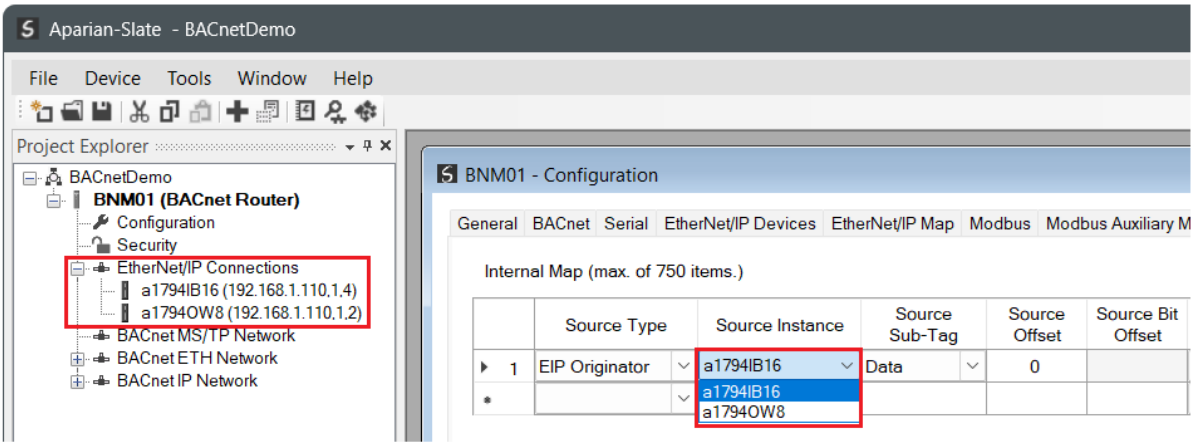


Figure 3.90 – IDS Copy – EtherNet/IP Originator Source Instance

The Source Offset is the offset in the selected EtherNet/IP device Class 1 **Input** Assembly. The Count is the number of **bytes** that will be copied.

The user can select to copy the data from the EtherNet/IP connection or the status.

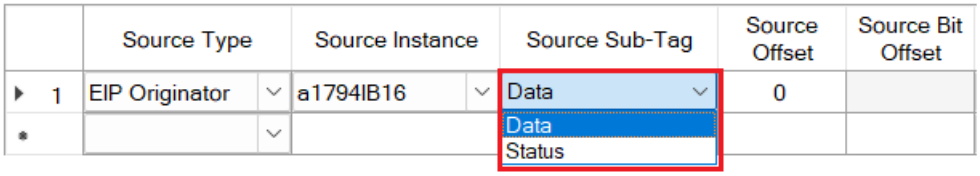


Figure 3.91 – IDS Copy – EtherNet/IP Originator Status

When selecting the status, the format of the Status information is shown below:

Parameter	Data Type	Description
EtherNet/IP Originator Connection Status	DINT	Bit 0 – Connection Ok

Table 3.14 – EtherNet/IP Originator Connection Status

3.9.1.4. MODBUS REGISTER

When copying Modbus data to the BACnet interface, the source type needs to be MB Register.

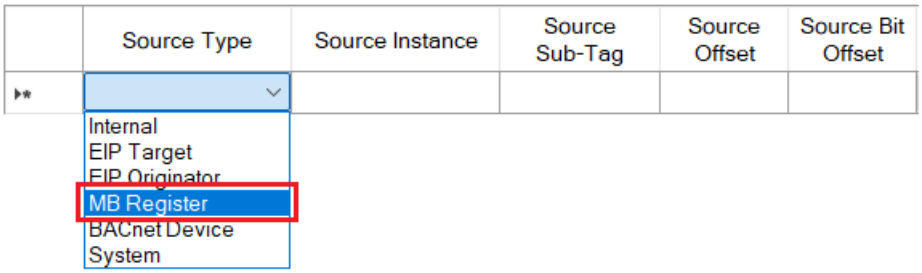


Figure 3.92 – IDS Copy - Modbus Source Type

The source instance will be the Modbus register type required.

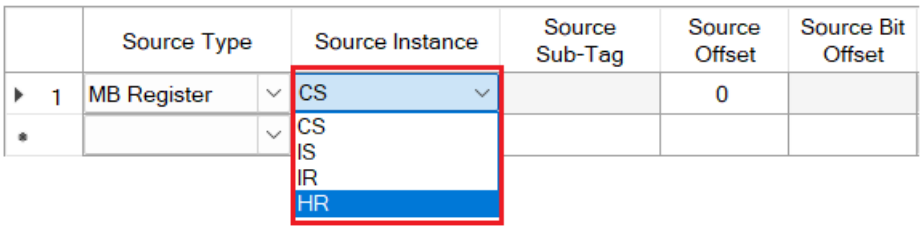


Figure 3.93 – IDS Copy - Modbus Source Instance

The Source Offset is the Modbus Register offset from where the data must be copied. The Count is the number of **bytes** that will be copied.

3.9.1.5. BACNET DEVICE

When copying data from a BACnet device, the ***Source Type*** needs to be ***BACnet Device***.

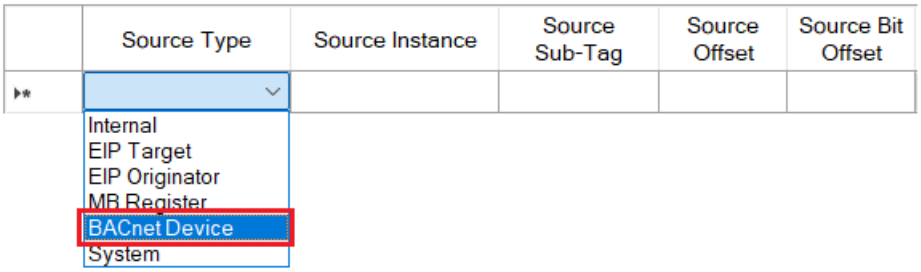


Figure 3.94 – IDS Copy – BACnet Device Source Type

The ***Source Instance*** will be the instance name of the configured BACnet device. The Source Sub-Tag will be Data or Status.

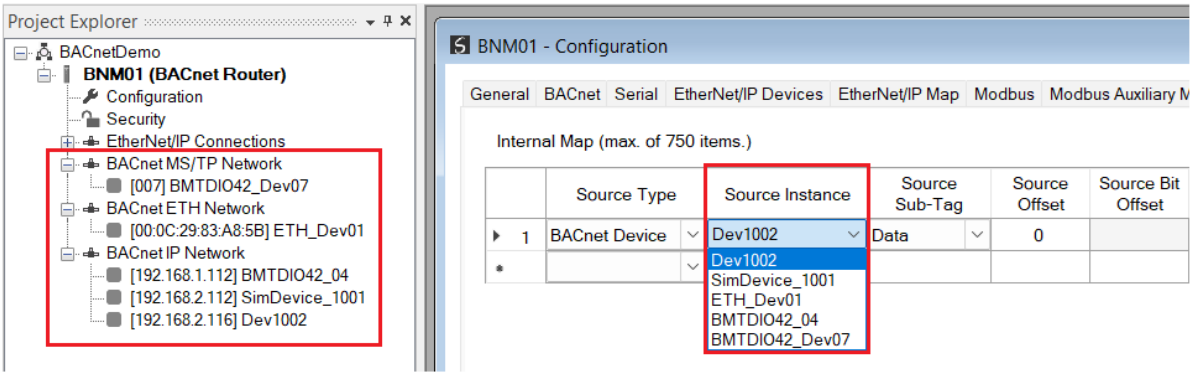


Figure 3.95 – IDS Copy – BACnet Source Instance

3.9.1.6. SYSTEM

When copying system information, the source type needs to be System.

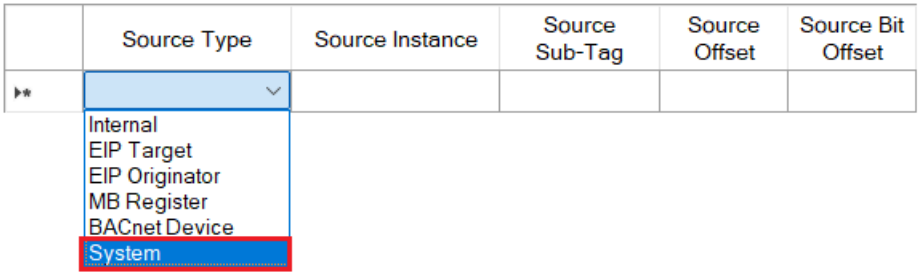


Figure 3.96 – IDS Copy – System Information

NOTE: The data structure is described in the Data Formats section in chapter 7.

3.9.2. COPY TO

One of six destinations can be selected to copy to:
Internal, EIP Target, EIP Originator, MB Register, BACnet Device or System.

3.9.2.1. INTERNAL

When copying data to the internal data space (IDS), the ***Destination Type*** needs to be Internal.

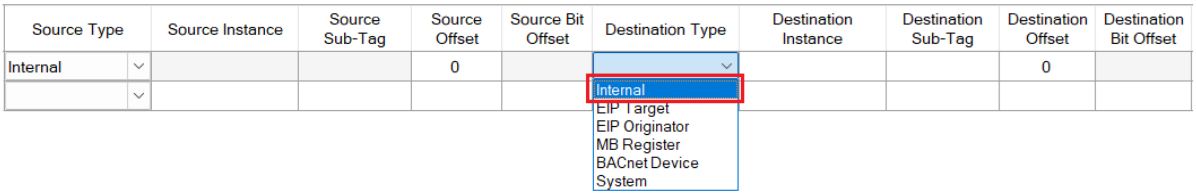


Figure 3.97 – IDS Copy – Internal Source Type

The ***Destination Instance*** is Not Applicable for the internal data space. The ***Destination Offset*** is the offset in the *Internal Data Space (IDS)* which has a maximum of 100 Kbytes. The ***Count*** is the number of **bytes** that will be copied.

3.9.2.2. EIP TARGET

When copying data to the EtherNet/IP Target input assembly, the destination type needs to be EIP Target.

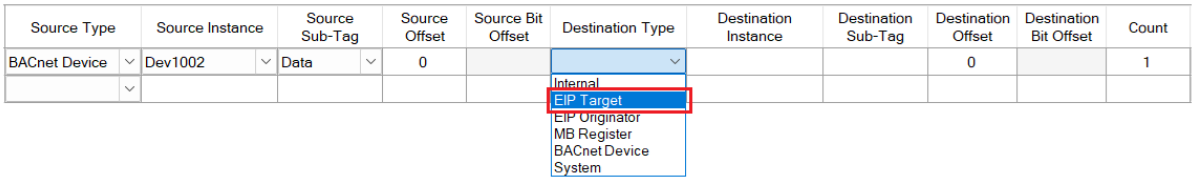


Figure 3.98 – IDS Copy – EtherNet/IP Target Destination Type

The ***Destination Instance*** will be the connection number, which can be connection 1 to 25, based on the number of connections configured. The ***Destination Offset*** is the offset of the EtherNet/IP input assembly to where the data must be copied. The ***Count*** is the number of **bytes** that will be copied.

3.9.2.3. EIP ORIGINATOR

When copying data to an EtherNet/IP IO device **Output** Assembly, the ***Destination Type*** needs to be EIP Originator.

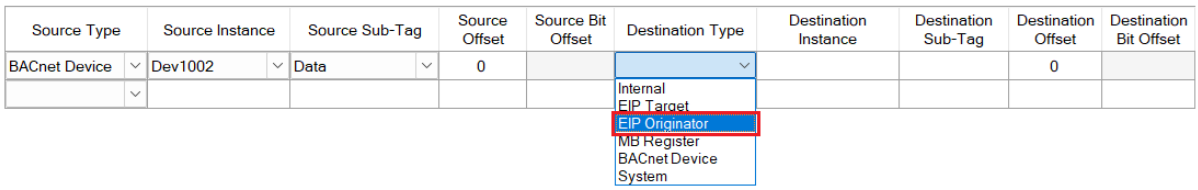


Figure 3.99 – IDS Copy – EtherNet/IP Originator Destination Type

The ***Destination Instance*** will be one of the EtherNet/IP IO devices added to the EtherNet/IP IO tree in Slate.

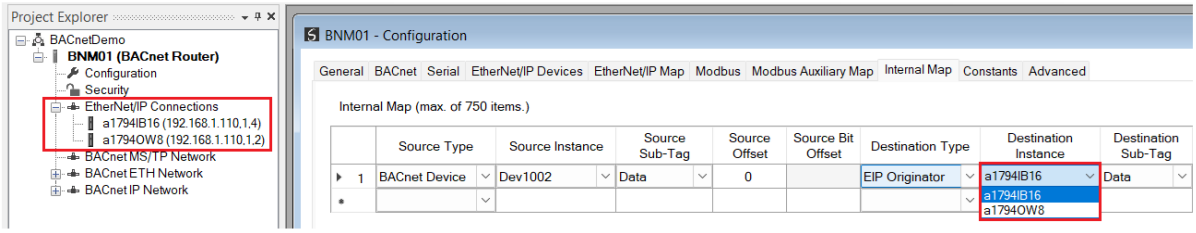


Figure 3.100 – IDS Copy – EtherNet/IP Originator Destination Instance

The ***Destination Offset*** is the offset in the selected EtherNet/IP device Class 1 **Output** Assembly. The Count is the number of **bytes** that will be copied.

3.9.2.4. MODBUS REGISTER

When copying data to a Modbus Register, the ***Destination Type*** needs to be MB Register.

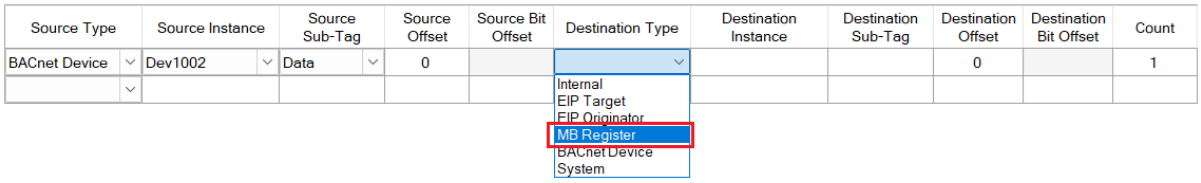


Figure 3.101 – IDS Copy - Modbus Destination Type

The ***Destination Instance*** will be the Modbus register type required.

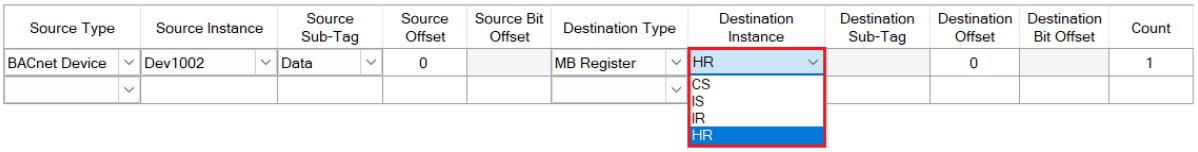


Figure 3.102 – IDS Copy - Modbus Destination Instance

The ***Destination Offset*** is the Modbus Register offset to where the data must be copied. The ***Count*** is the number of **bytes** that will be copied.

3.9.2.5. BACNET DEVICE

When copying data to a BACnet device, the ***Destination Type*** needs to be BACnet Device.

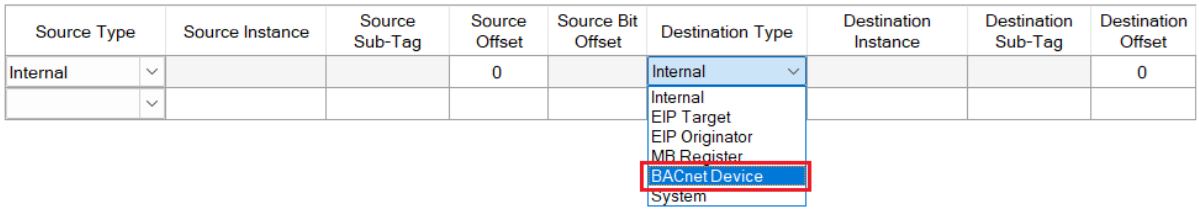


Figure 3.103 – IDS Copy – BACnet Destination Type

The ***Destination Instance*** will be the configured instance name of the BACnet device.

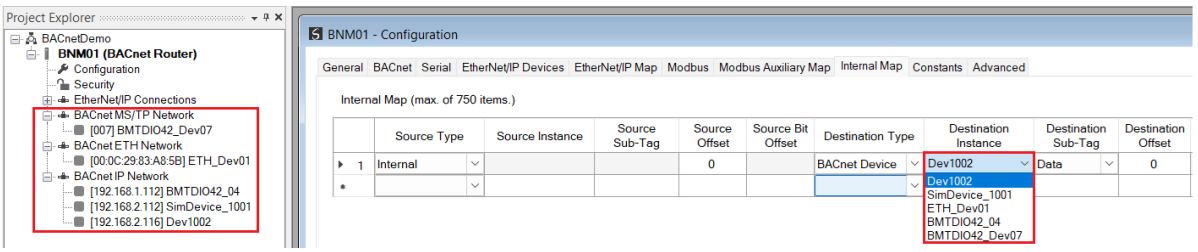


Figure 3.104 – IDS Copy – BACnet Destination Instance

The ***Destination Offset*** is the offset in the BACnet data to where the data must be copied. The ***Count*** is the number of **bytes** that will be copied.

3.9.2.6. SYSTEM

When copying data to the System (Control), the ***Destination Type*** needs to be System.

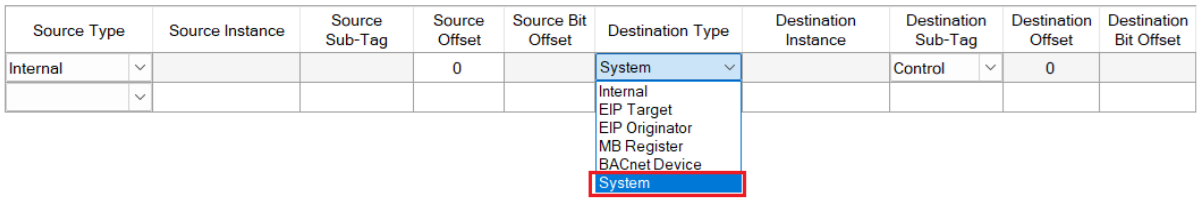


Figure 3.105 – IDS Copy – System Destination Instance

The ***Destination Instance*** is Not Applicable.
The ***Destination Sub-Tag*** will be ***Control***.
The ***Destination Offset*** is the offset in the System Control structure to where the data must be copied.
The ***Count*** is the number of **bytes** that will be copied.

NOTE: The data structure is described in the Data Formats section in chapter 7.

NOTE: For more information regarding the specific Internal Map operation for specific interfaces, see the setup and configuration sections for the various BACnet and Primary Interfaces.

3.10. CONSTANTS

The Internal Data Constant configuration provides a mechanism to load the Internal Data Space (IDS) with constant data at module start-up. This is often useful for the pre-population of configuration data for BACnet, Modbus and EtherNet/IP devices.

The ***Constants*** configuration window is opened by either double clicking on the module in the tree, or by right-clicking the module and selecting *Configuration*.

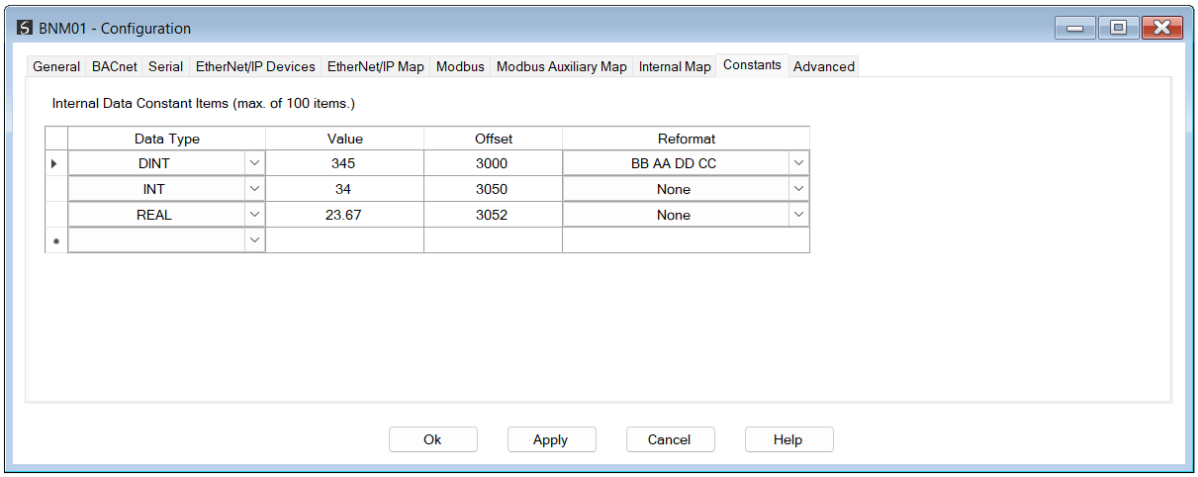


Figure 3.106 – Constants configuration

The ***Constant*** configuration consists of the following parameters:

Parameter	Description
Data Type	The Data Type of the constant item, either: • SINT • INT • DINT • LINT • REAL • USINT • UINT • UDINT
Value	The constant value.
Offset	The offset in the Internal Data Space where the constant value will be copied.
Reformat	Used to specify how the data is formatted before writing to Internal Data Space: • **None** – No reformatting applied. (AA BB or AA BB CC DD). • **BB AA** – 16bit Byte swap • **BB AA DD CC** – 32bit Byte Pair Swap • **CC DD AA BB** – Word Swap • **DD CC BB AA** – Word and Byte Pair Swap

Table 3.15 – Constants configuration parameters

3.11. ADVANCED

The Advanced configuration window is opened by either double clicking on the module in the tree, or by right-clicking the module and selecting *Configuration*.

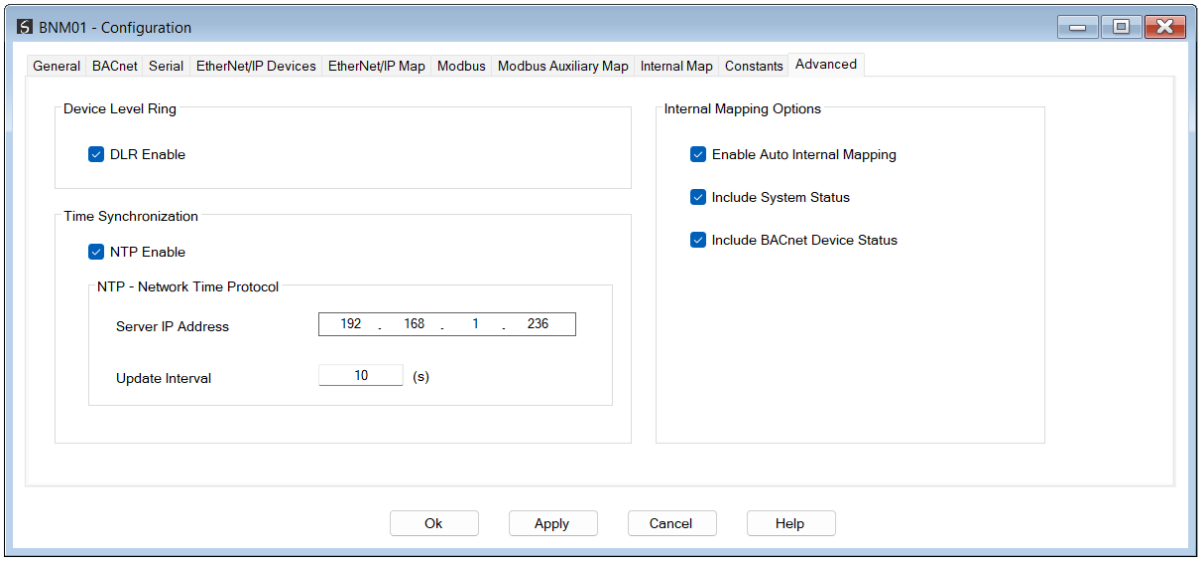


Figure 3.107 – Advanced configuration

The Advanced configuration consists of the following parameters:

Parameter	Description
Device Level Ring	
DLR Enable	Enable the Ethernet Device Level Ring functionality.
Time Synchronization	
NTP Enable	The BACnet Master can synchronize its onboard clock to an NTP Server by enabling NTP.
NTP – Server IP Address	This setting is the IP address of the NTP Server which will be used as a time source.
NTP – Update Interval	This setting is the updated interval (in seconds) that the BACnet Master will request time from the NTP Server.
Internal Mapping Options	
Enable Auto Internal Mapping	When selected, the Internal Mapping will be automatically created and the manually editing of the internal mapping will not be permitted.
Include System Status	Specifies whether the module System Status information should be included in the internal mapping.
Include BACnet Device Status	Specifies whether the device status for each BACnet device should be included in the internal mapping.

Table 3.16 – Advanced configuration parameters

3.12. MODULE DOWNLOAD

Once the BACnet Master configuration has been completed, it must be downloaded to the module. Before downloading the ***Connection Path*** of the module should be set. This path will automatically default to the IP address of the module, as set in the module configuration. It can however be modified, if the BACnet Master is not on a local network.

The connection path can be set by right-clicking on the module and selecting the ***Connection Path*** option.

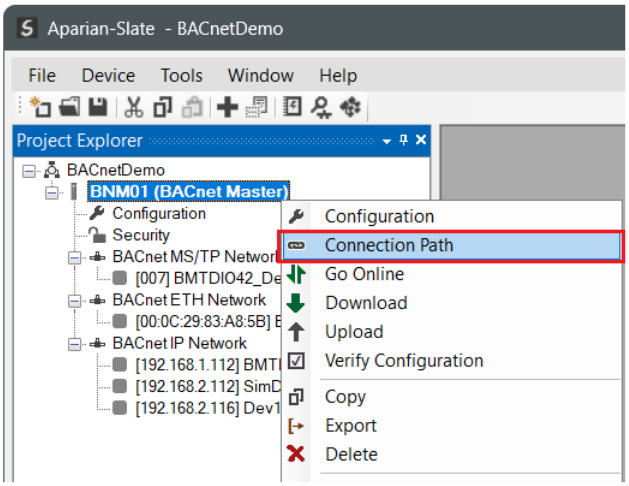


Figure 3.108 - Selecting Connection Path

The new connection path can then be either entered manually or selected by means of the ***Target Browser***.

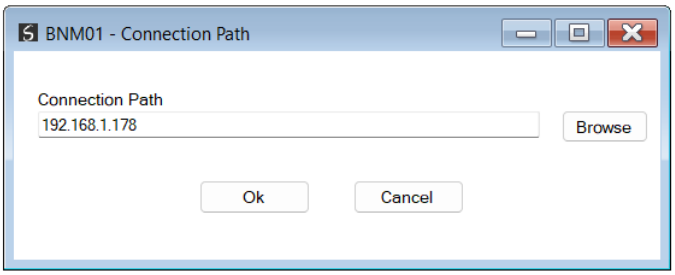


Figure 3.109 - Connection Path

To initiate the download, right-click on the module and select the ***Download*** option.

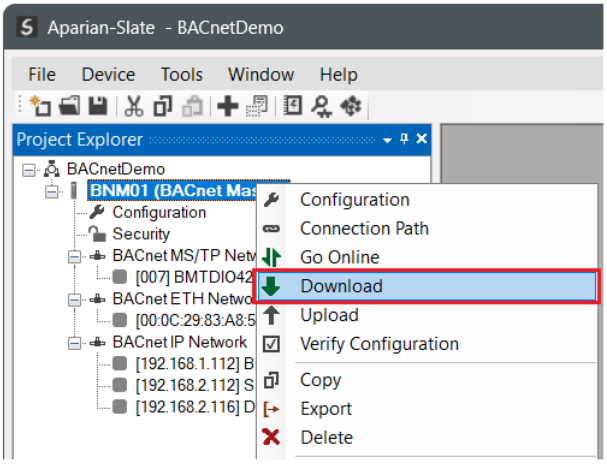


Figure 3.110 - Selecting Download

Once complete, the user will be notified that the download was successful.

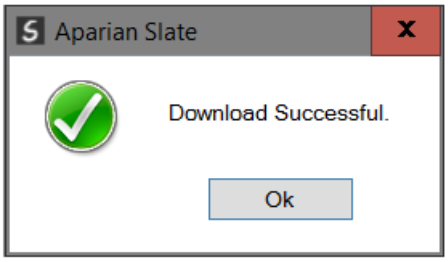


Figure 3.111 - Successful download

Within the Slate environment the module will be in the Online state, indicated by the green circle around the module. The module is now configured and will start operating immediately.

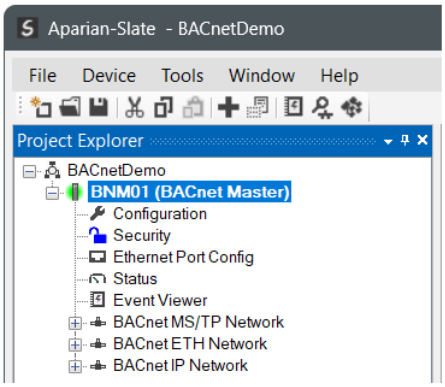


Figure 3.112 - Module online

4. SD CARD

The BACnet Master supports an SD Card (see below) which can be used for disaster recovery. The SD Card can be pre-loaded with the required firmware, network configuration, and/or application configuration.

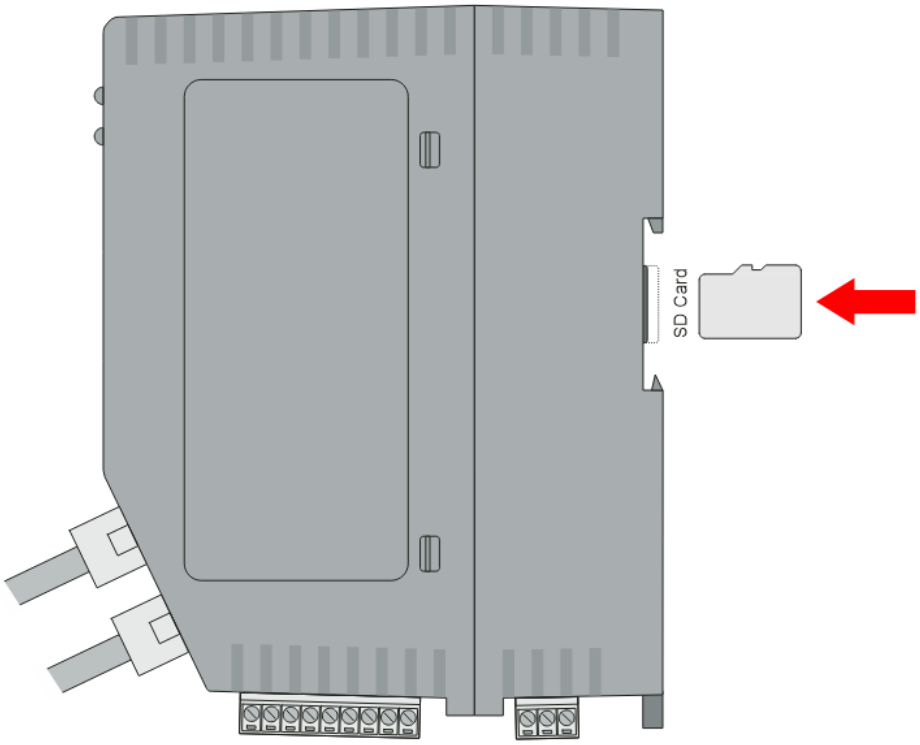


Figure 4.1 – Module Side View – SD Card Slot

NOTE: The user will need to ensure that the SD Card has been formatted for FAT32.

NOTE: All needed files must be copied into the root directory of the SD Card. The module will not use files which are located in folders.

4.1. FIRMWARE

The user can copy the required firmware (which can be downloaded from the Aparian website) onto the root directory of the SD Card.

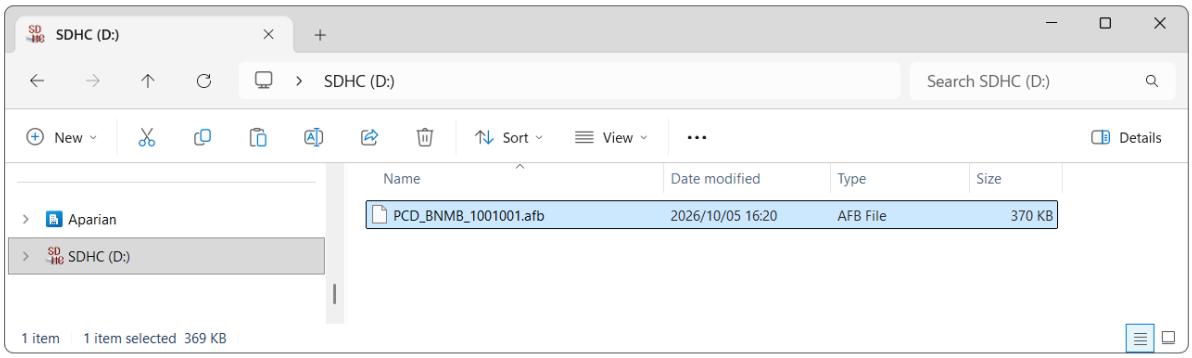


Figure 4.2 – SD Card – Firmware file

NOTE: The filename of the firmware file must not be changed. The specific module will use only the firmware that is valid (e.g. the BACnet Master Base will only use the "PCD_BNMB_xxxxxxx.afb" firmware file).

NOTE: If more than one firmware file, with different firmware revisions, of the same product is on the SD Card it can cause the module to constantly firmware upgrade the module.

If a faulty module is replaced the user can insert the SD Card with the firmware file on into the new module. While the module is booting it will detect if the firmware on the new module is different from that on the SD Card. If yes, the firmware will either be upgraded or downgraded to the firmware revision on the SD Card.

4.2. NETWORK PARAMETERS

When the SD Card has been inserted into the module and the user is online with the module in Slate, then the user has the option to directly upload the network parameters (e.g., IP Address, Subnet Mask, etc.) on to the SD Card using the *Save Network Parameters to SD Card* option. This will copy the module's current network parameters to the SD Card.

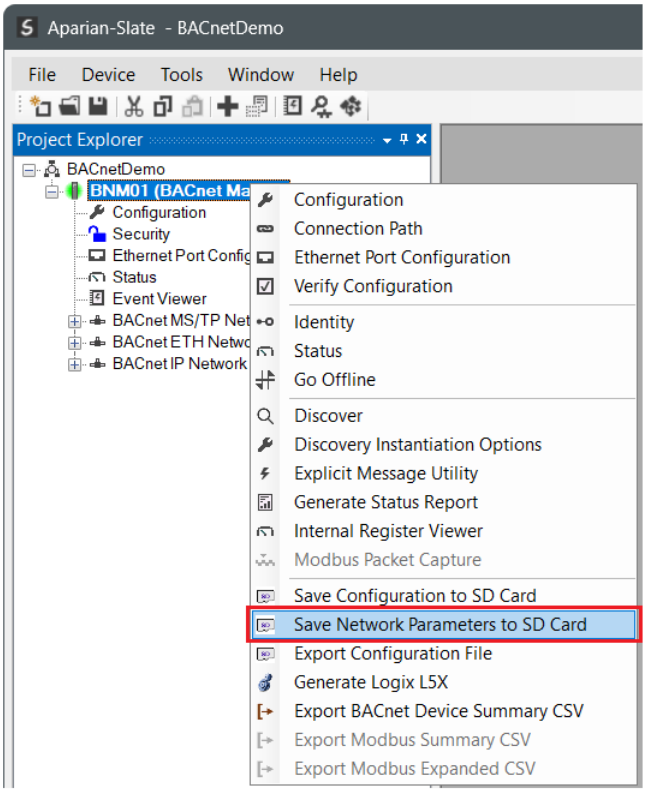


Figure 4.3 – SD Card – Network Parameters

When the module powers up with an SD Card inserted and there exist network parameters saved onto the SD Card, it will update the existing network parameters if they are different.

4.3. CONFIGURATION

If a module needs to be replaced, the user can insert the SD Card (with the configuration file) into the new module. The new module will determine if the configuration on the SD Card is different than the currently loaded configuration (even when there is no configuration on the module). If different, the configuration on the SD Card will be downloaded into the module's NV memory before the module starts executing.

The user can add the Slate configuration file to the SD Card root directory in one of two ways.

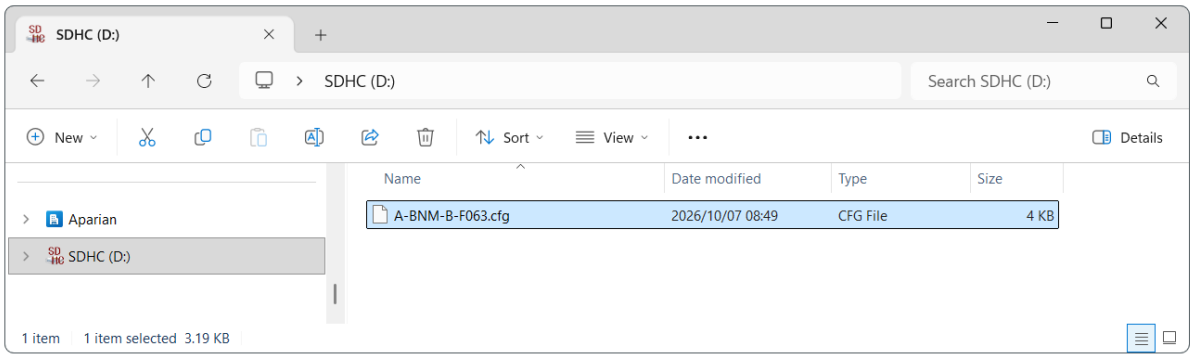


Figure 4.4 – SD Card – Configuration file

4.3.1. Manual Copy

Once the user has created the application configuration in the Slate, the configuration can be exported to a file that can be used on the SD Card. Once the file has been created the user can copy this file into the root directory of the SD Card.

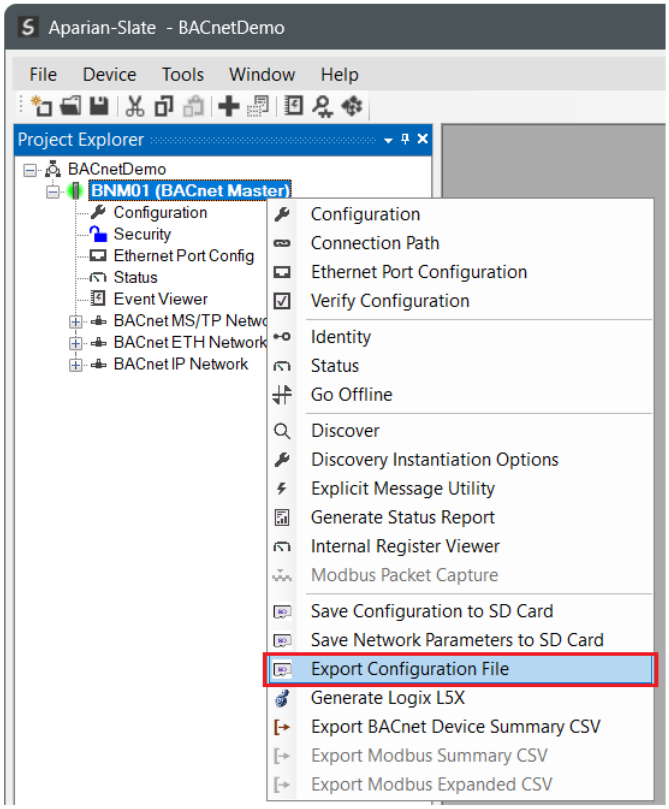


Figure 4.5 – Configuration Export for SD Card

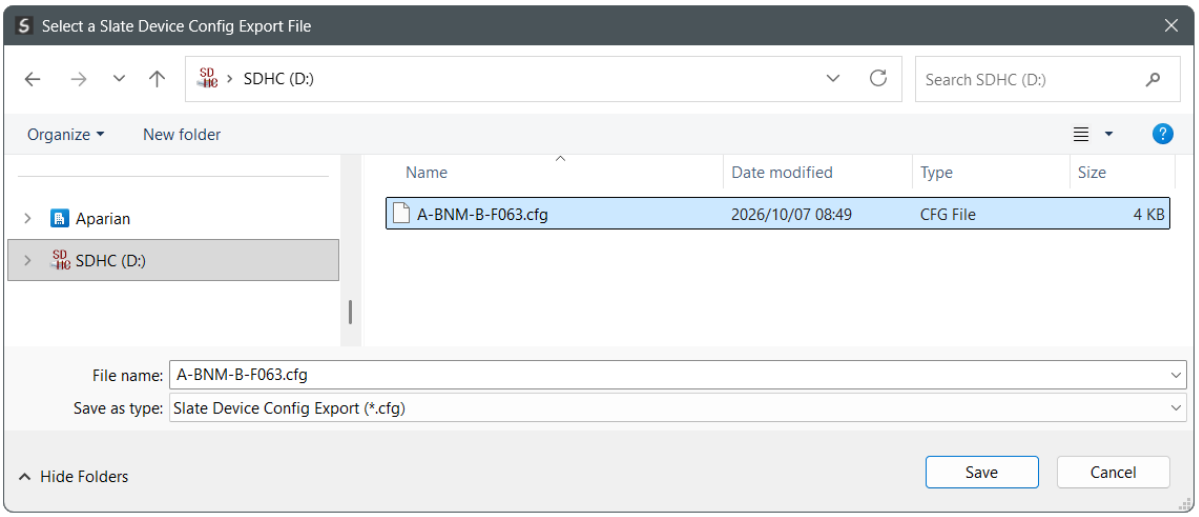


Figure 4.6 – Configuration Export for SD Card

NOTE: The filename of the configuration file must **not** be changed. The specific module will use only the configuration that is valid (e.g. the BACnet Master will only use the BACnet configuration file).

NOTE: If more than one configuration file, with different configuration signatures, of the same product is on the SD Card then only the last configuration will be used.

4.3.2. SLATE TRIGGERED UPLOAD

When the SD Card has been inserted into the module and the user is online with the module in Slate, then the user has the option to directly upload the configuration on to the SD Card using the *Save Configuration to SD Card* option. This will copy the configuration that has been downloaded to the module directly to the SD Card without the need to remove it from the module and inserted into a PC.

NOTE: All other configuration files in the SD Card root directory will be deleted when the upload is done.

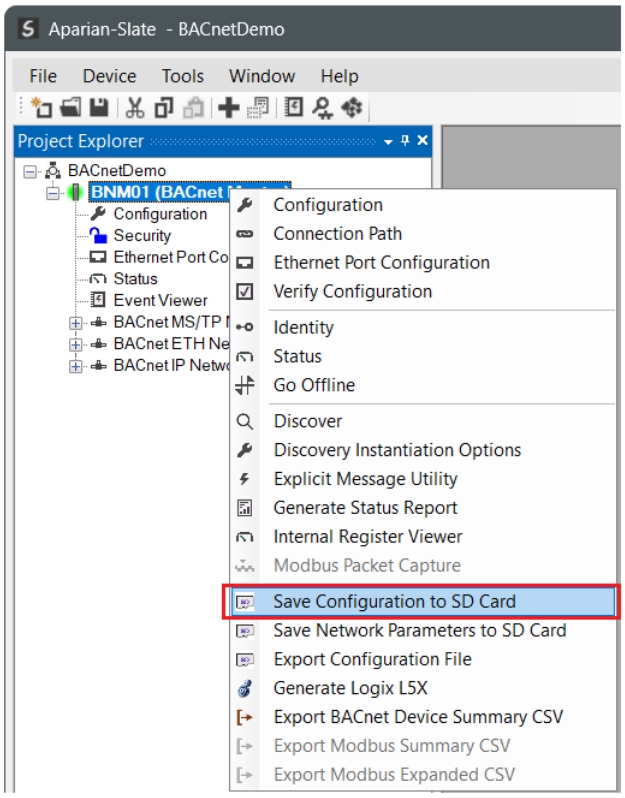


Figure 4.7 – Save Configuration to SD Card

5. DEVICE FIRMWARE UPDATE

The BACnet Master module supports in-field firmware upgrading. The latest firmware for the module can be downloaded from the Aparian website ***www.aparian.com***. The firmware is digitally signed, so only the correct firmware can be used.

To firmware upgrade the module, follow the steps below:

- From the ***Tools*** menu in Slate, select the ***DeviceFlash*** utility.

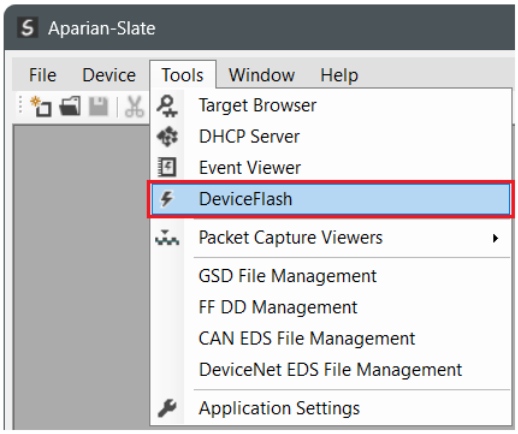


Figure 5.1 – Select DeviceFlash utility from Slate

- When the utility opens, the user will be prompted to select the binary file to be used to firmware upgrade the module.

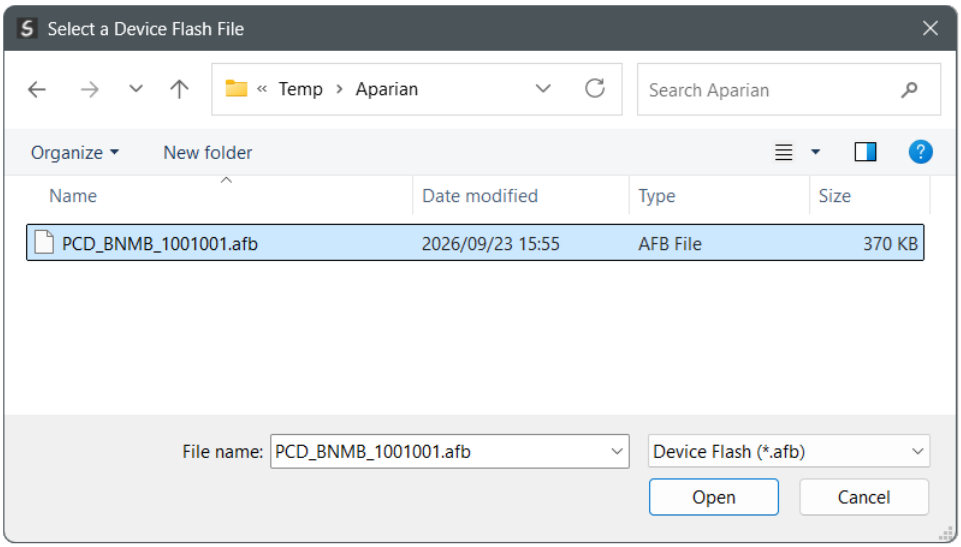


Figure 5.2 – Select the binary file

- After selecting the file, the user will be prompted to select the device to firmware upgrade on the local network.

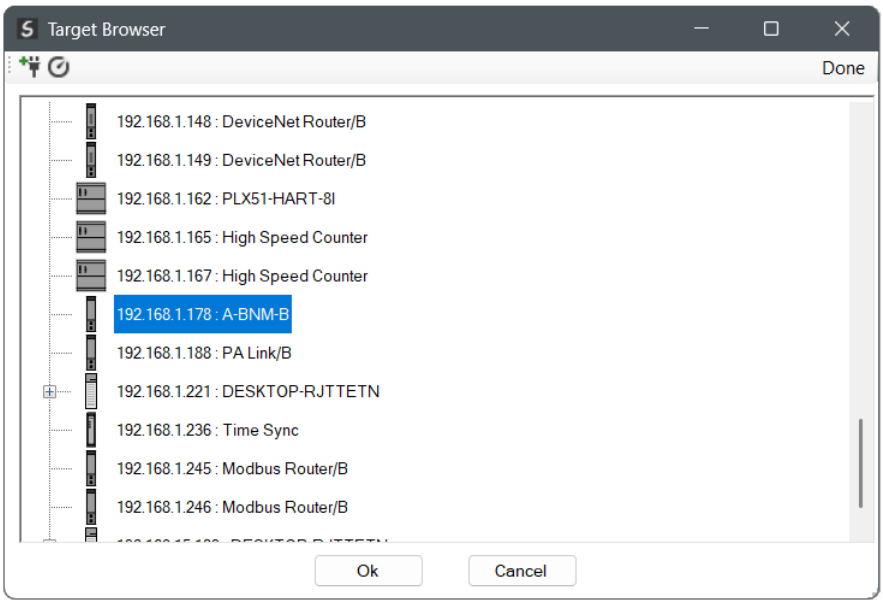


Figure 5.3 – Select the device to be updated

- After the device selection, the user will be prompted to start the device flash process. The firmware update will take less than 2 minutes to complete.

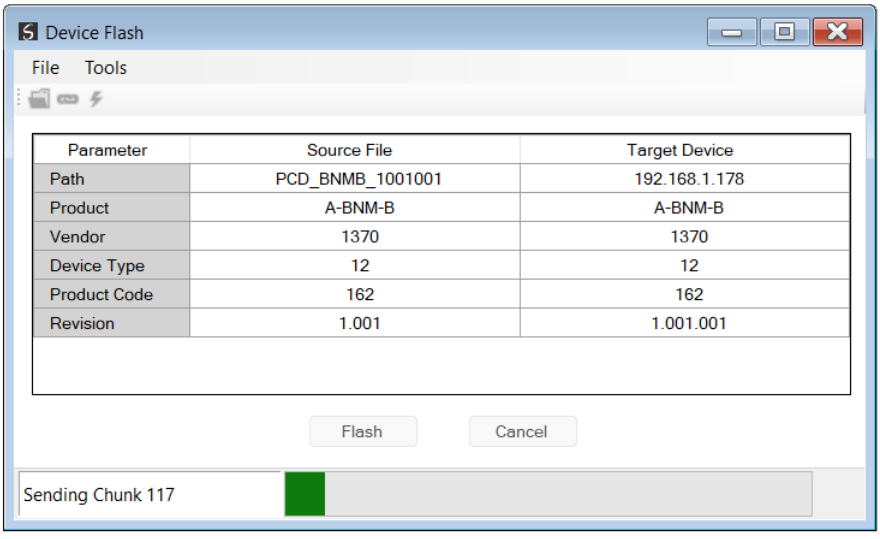


Figure 5.4 – Firmware update busy

- Once the firmware update has successfully completed, the Target Device textboxes will display green.

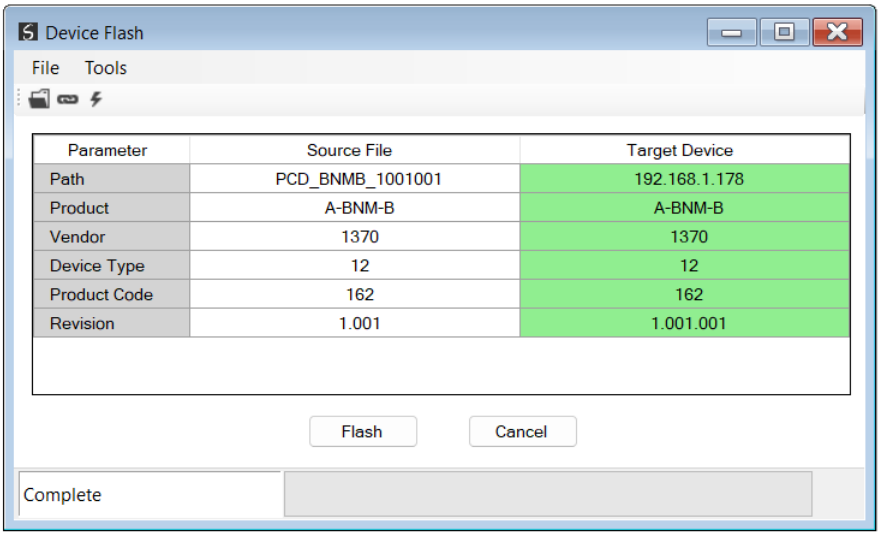


Figure 5.5 – Firmware update successfully completed

NOTE: If for any reason the firmware update failed (e.g. power down during the update), then the module will revert back to the bootloader. The user can then simply reflash the module again to update it to the latest application firmware.

6. SECURITY

The BACnet Master supports security services allowing the user to configure various levels of module security.

When security is enabled all EtherNet/IP communication between Slate and the module is encrypted preventing information from being intercepted by a third party.

Security of the module is configured and downloaded separately to the module's primary (application) configuration and can thus be configured online without affecting the running process.

The security services provide the following features:

- Configurable level of security
- Encrypted EtherNet/IP communication
- Fixed User Roles and Custom Users
- Configurable Password rules
- Login Expiry
- Failed Login Cool-off and Account Disable options
- Configurable Global Services and Custom Ports
- Optional IP Address Access Control List
- Optional MAC Access Control List
- Restrictions on Reset command
- Restrictions on Device Flash options
- Security Audit trail in Event Log

Optional user authentication can be enabled, requiring users to login to the module when connecting. In addition to the fixed role users, (viz. Administrator, Engineer, Operator, Auditor and Viewer) up to 5 custom users can be configured. Each custom user is assigned one of the aforementioned fixed roles.

Once user authentication has been initialized, only users with Administrator privileges are able to change the security configuration and reset passwords etc.

Associated privileges for each user role, is tabulated below:

Privileges / Roles	Administrator	Engineer	Operator	Auditor	Viewer	None
System						
Identity Object	✓	✓	✓	✓	✓	✓
Device Flash (Firmware Upgrade)	✓	-	-	-	-	-
Reset Module	✓	✓	-	-	-	-
Change Ethernet Interface Settings	✓	✓	-	-	-	-
Application						
Download/Upload Configuration	✓	✓	-	-	-	-
View Status and Statistics	✓	✓	✓	✓	✓	-
Clear Statistics	✓	✓	✓	-	-	-
Packet Capture	✓	✓	✓	✓	-	-
View Event Log	✓	✓	✓	✓	✓	-
Clear Event Log	✓	-	-	-	-	-
Generate Status Report	✓	✓	✓	✓	-	-
Modbus Pass Through Functions	✓	✓	-	-	-	-
Internal Register Viewer	✓	✓	-	-	-	-
Security						
Download/Upload Security Configuration	✓	-	-	-	-	-
Clear All Security	✓	-	-	-	-	-
View Security Status and Statistics	✓	✓	-	✓	-	-
Clear Security Statistics	✓	-	-	-	-	-
Change Own Password	✓	✓	✓	✓	✓	-
Change Another User's Password	✓	-	-	-	-	-

Table 6.1 – User Role Privileges

6.1. SECURING PROCESS

The process to secure a module involves the following basic steps:

- Setup Security **Configuration**
- **Download** Security Configuration to the module
- **Initialize** User Authentication

6.1.1. CONFIGURATION

The Security Configuration is accessed by right-clicking on the ***Security*** item in the device tree and selecting the ***Security Configuration*** option.

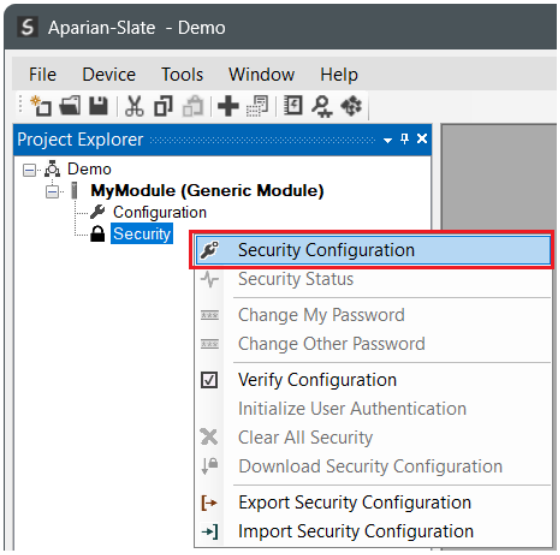


Figure 6.1 – Selecting Security Configuration

The configuration form will open and display multiple configuration tabs.

6.1.1.1. GENERAL CONFIGURATION

The ***General*** configuration tab is shown below.

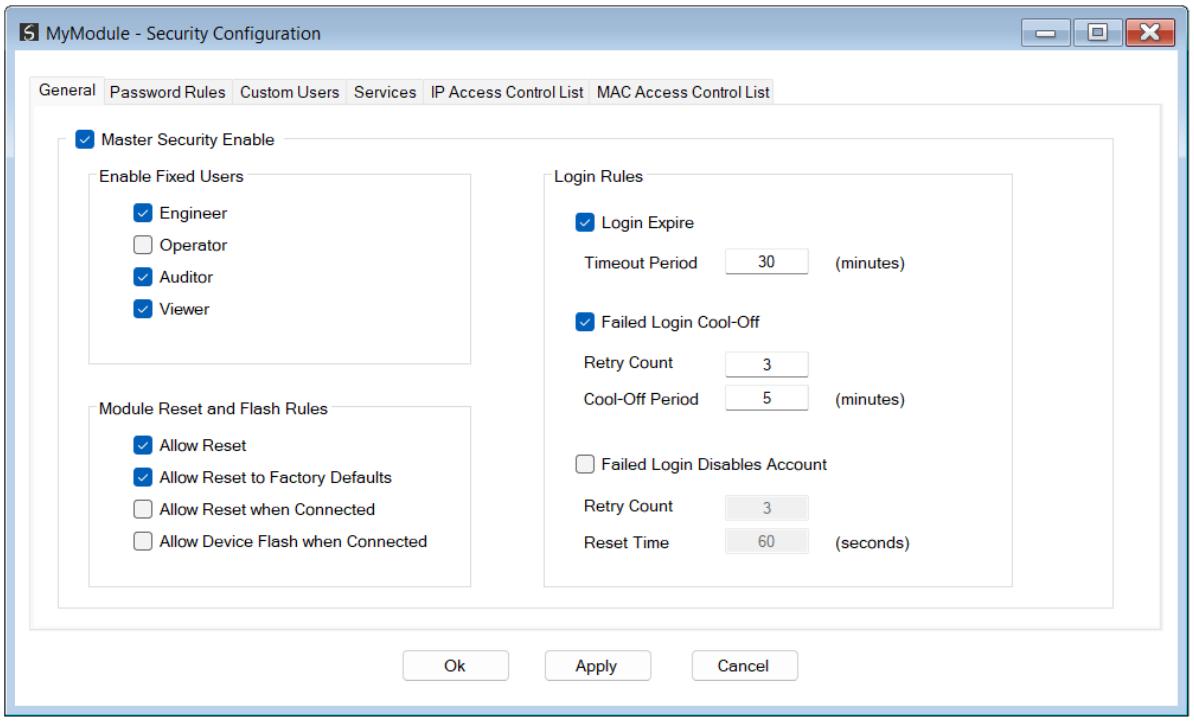


Figure 6.2 – Security Configuration – General

The ***General*** configuration tab comprises the following parameters:

Parameter	Description
Master Security Enable	Used to enable or disable the master security.  **NOTE**: If this option is disabled then all Security measures are disabled.
Enable Fixed Users ***Engineer*** ***Operator*** ***Auditor*** ***Viewer***	Allows each of the fixed users to be enabled or disabled.  **NOTE**: The fixed ***Administrator*** user cannot be disabled.
Module Reset and Device Flash Rules	
Allow Reset	Must be checked to allow any type of Reset via EtherNet/IP.
Allow Reset to Factory Defaults	Must be checked to allow Reset to Factory Defaults (including Reset to Factory Defaults except Communication settings).
Allow Reset when Connected	Must be checked to allow any type of Reset when the module is connected, where connected implies an EtherNet/IP Class 1 connection.
Allow Device Flash when Connected	Must be checked to allow Device Flash when the module is connected, where connected implies an EtherNet/IP Class 1 connection.
Login Rules	
Login Expire	Selecting this option causes all users to be automatically logged-out after the Login Expire Timeout Period.
Login Expire Timeout Period	The duration, in minutes, after which all users will be automatically logged-out. Min: 1 minute Max: 1440 minutes (24 hours)
Failed Login Cool-Off	Selecting this option initiates a Cool-off period after a user has entered an incorrect password the Retry Count times in succession. During the cool off period the user will not be able to login.  **NOTE**: This option, and the Failed Login Disable Account option, cannot be simultaneously selected.
Failed Login Cool-Off Retry Count	The number of retries before the Login Cool-Off period is applied.
Failed Login Cool-Off Period	The Cool-Off duration, in minutes. Once a user has entered an incorrect password the Retry Count times in succession, they would need to wait this period before they can login again. Min: 1 minute Max: 1440 minutes (24 hours)

Failed Login Disables Account	Selecting this option causes the user's account to be disabled if they enter an incorrect password the Retry Count times in succession before the Reset Time has expired. **NOTE**: Once the user's account has been disabled, an Administrator would need to reset their password. **NOTE**: This option will not be applicable for the Administrator user account.
Failed Login Disables Account Retry Count	The number of retries while the Reset Time is active before the user's account is disabled.
Failed Login Disables Account Reset Time	The Reset Time duration, in seconds. Once a user has entered an incorrect password, the Disable Account Reset Time will be applied and starts counting down to zero. If the user enters the incorrect Password the Retry Count times before the Reset Time has expired, the user's account will be disabled. Once the Reset Time has expired (gone back to zero), then the login process can be restarted. Min: 1 second. Max: 65000 seconds (18 hours)

Table 6.2 – Security Configuration – General

6.1.1.2. PASSWORD RULE CONFIGURATION

The ***Password Rules*** tab is shown below.

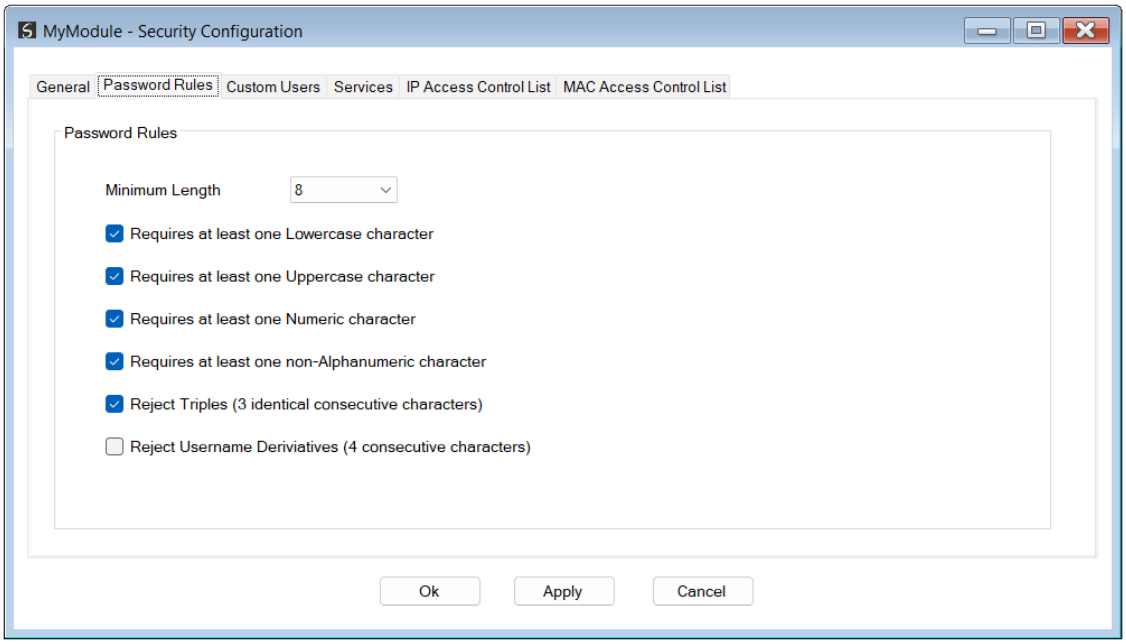


Figure 6.3 – Security Configuration – Password Rules

The ***Password Rules*** tab comprises the following parameters:

Parameter	Description
Minimum Length	The minimum number of characters the password must contain. Min: 1 Max: 32
Requires at least one **Lowercase** character	When selected, all passwords will require at least one lowercase character. e.g. a b c d
Requires at least one **Uppercase** character	When selected, all passwords will require at least one uppercase character. e.g. A B C D
Requires at least one **Numeric** character	When selected, all passwords will require at least one non-alphanumeric character. e.g. 1 2 3 4
Requires at least one **Non-Alphanumeric** character	When selected, all passwords will require at least one non-alphanumeric character. e.g. ! @ # $ %
Reject **Triples**	When selected, all passwords may not contain 3 identical (case-insensitive) consecutive characters. e.g. MyDDDog, strangecCcat.
Reject Username Derivative	When selected, all passwords may not contain any derivative of the respective username. A derivative is defined as any 4 consecutive characters of the password case-insensitive-matching any 4 consecutive characters of the username.

Table 6.3 – Security Configuration – Password Rules

6.1.1.3. CUSTOM USERS CONFIGURATION

The ***Custom Users*** configuration tab is shown below.

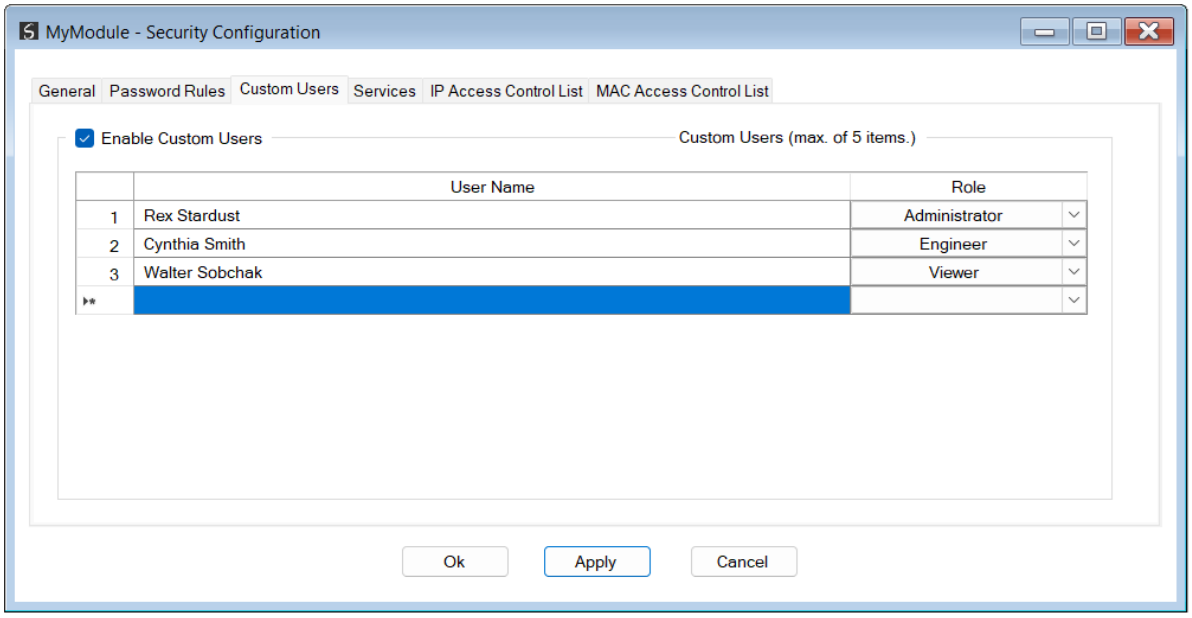


Figure 6.4 – Security Configuration – Custom Users

The ***Custom Users*** configuration tab comprises the following parameters:

Parameter	Description
Enable Custom Users	Global Enable / Disable for all Custom Users
User Name	User Name of the Custom User. **NOTE**: The User Name cannot exceed 32 characters. **NOTE**: The User Name cannot be the same as any of the fixed roles. (Administrator, Engineer etc.)
Role	The role allocated to this user. Select one of the following: Administrator Engineer Operator Auditor Viewer

Table 6.4 – Security Configuration – Custom Users

6.1.1.4. SERVICES CONFIGURATION

The ***Services*** configuration tab is shown below.

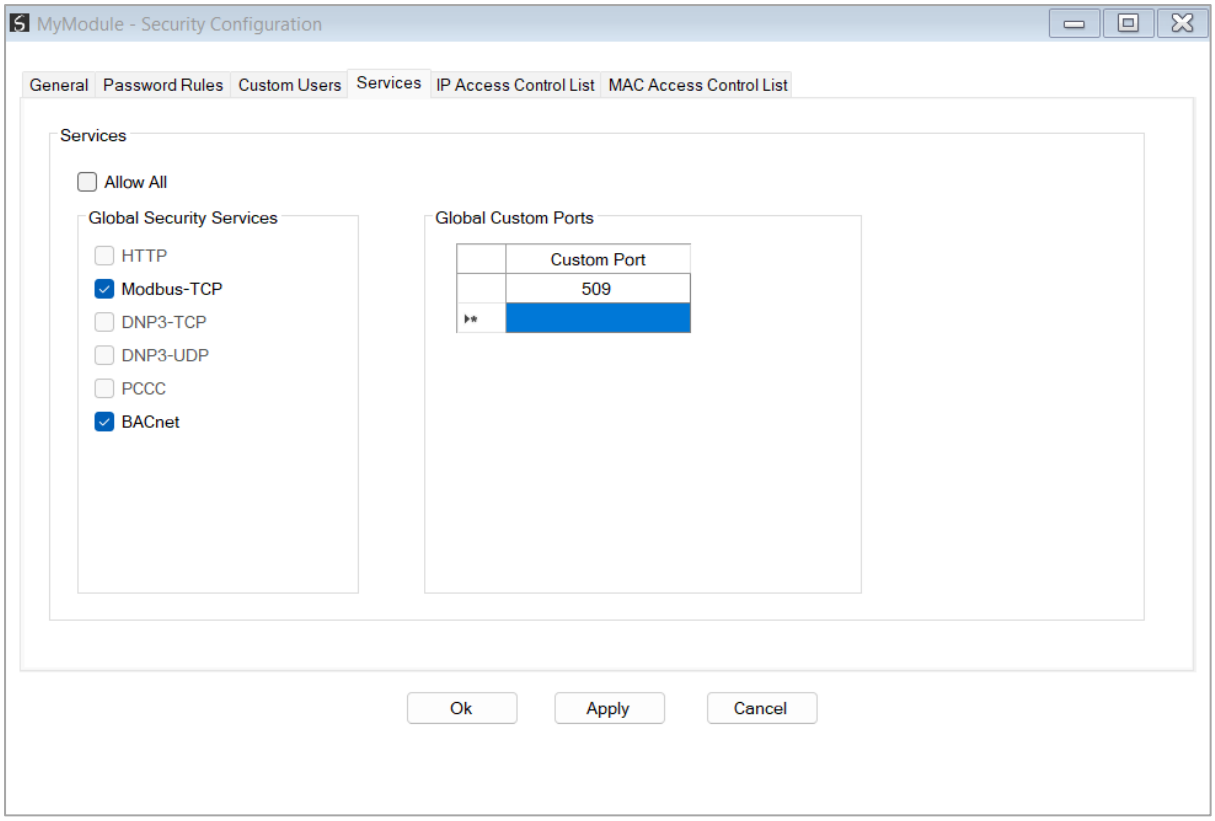


Figure 6.5 – Security Configuration – Services

The ***Services*** configuration tab comprises the following parameters:

Parameter	Description
Allow All	When selected all services are allowed by default.
HTTP	Allow HTTP web server connections to be accepted.
Modbus-TCP	Allow Modbus-TCP connections to be accepted.
DNP3-TCP	Allow DNP3 TCP connections to be accepted.
DNP3-UDP	Allow DNP3 UDP connections to be accepted.
PCCC	Allow PCCC connections to be accepted.
BACnet	Allow BACnet connections to be accepted.
Custom Ports	Up to 4 custom (TCP / UDP) ports can be added to the allowed list. **NOTE**: When the application is using non-standard ports for communication services (e.g. Modus-TCP) then these ports will need to be added to the Custom Ports list.

Table 6.5 – Security Configuration – Services

6.1.1.5. IP ACCESS CONTROL LIST CONFIGURATION

Enabling the IP Access Control List will block all communication from PCs (and other devices) unless their IP addresses are explicitly included in the list with the allowed services.

The ***IP Access Control List*** configuration tab is shown below.

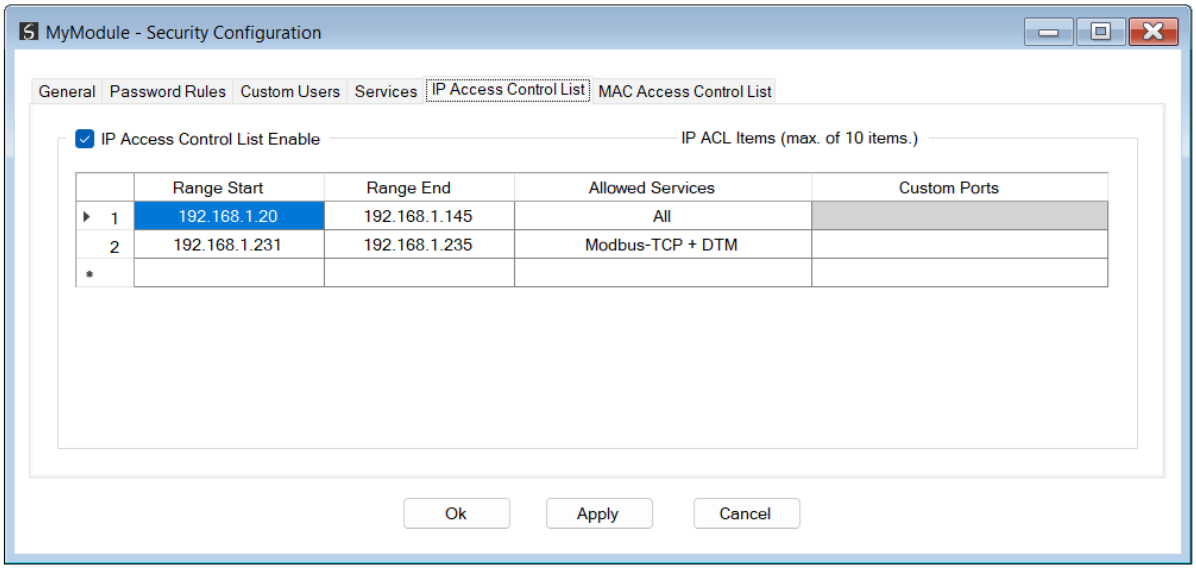


Figure 6.6 – Security Configuration – IP Access Control List

The ***IP Access Control List*** configuration tab comprises the following parameters:

Parameter	**Description**
IP Access Control List Enable	Enable / Disable for all IP Access Control List
IP Access Control List	
Range Start	First IP address in the range.
Range End	Last IP address in the range.
Allowed Services	The services allowed for this IP range.
Custom Ports	Additional custom ports allowed for this IP range.

Table 6.6 – Security Configuration – IP Access Control List.

To modify the Allowed Services the user can either click on the specific Allowed Service cell, or right-click on the specific row and select the ***Edit Services*** option.

The Service Selection form will open allowing the user to select each service's checkbox.

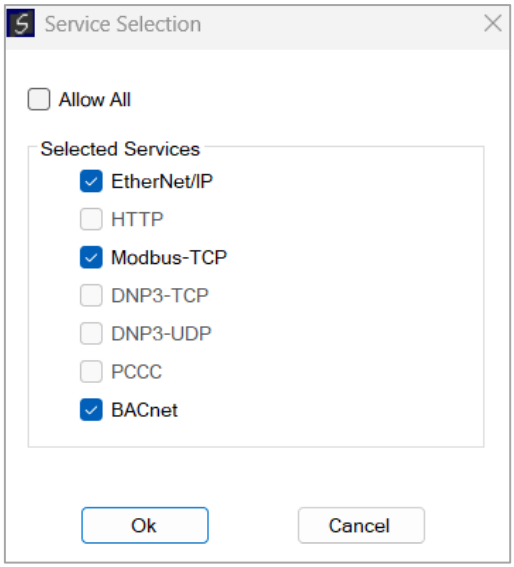


Figure 6.7 – Security Configuration – Service Selection

To modify the Custom Ports the user can either click on the specific ***Custom Ports*** cell, or right-click on the specific row and select the ***Edit Custom Ports*** option.

NOTE: The Custom Ports option is not available if the ***Allow All*** option has been selected in the Allowed Services.

The ***Custom Ports Selection*** form will open allowing the user to modify the Custom Ports.

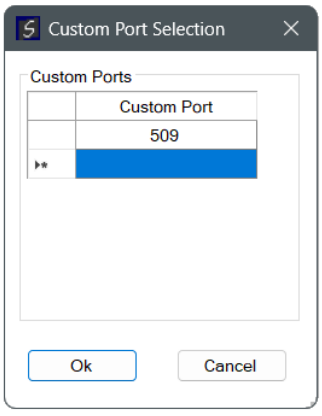


Figure 6.8 – Security Configuration – Custom Ports

6.1.1.6. MAC ACCESS CONTROL LIST CONFIGURATION

Enabling the MAC Access Control List will block all communication from PCs (and other devices) unless their MAC addresses are explicitly added to the list

NOTE: Incorrectly configuring the MAC Access Control List may result in the module no longer being able to communicate with Slate.

The ***MAC Access Control List*** configuration tab is shown below.

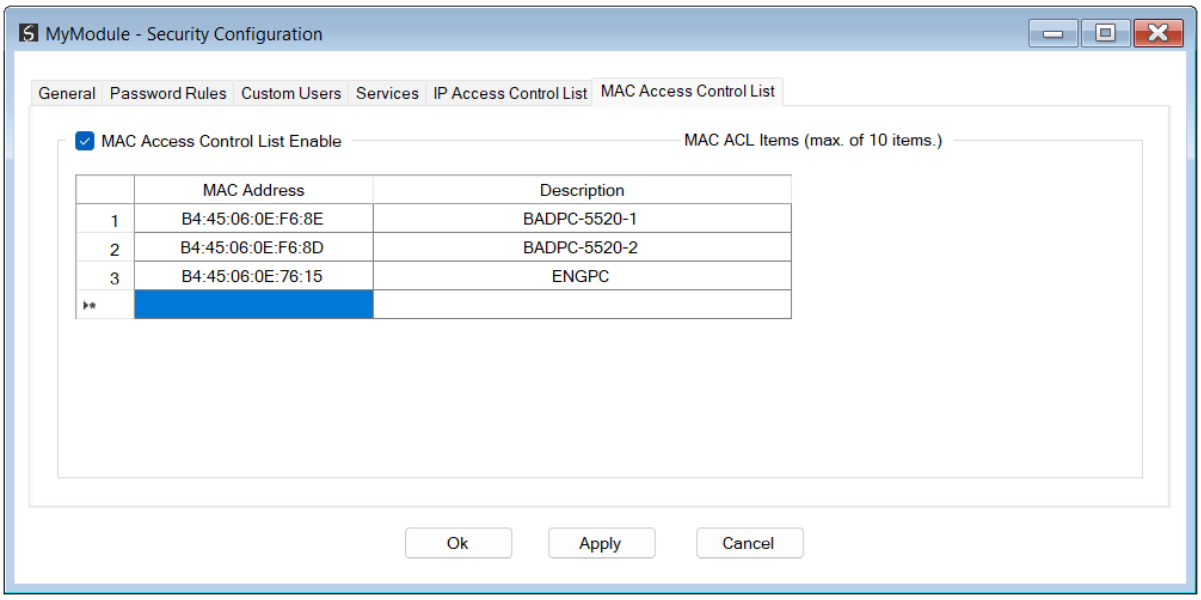


Figure 6.9 – Security Configuration – MAC Access Control List

The ***MAC Access Control List*** configuration tab comprises the following parameters:

Parameter	Description
MAC Access Control List Enable	Enable / Disable for all MAC Access Control List
MAC Access Control List	
MAC Address	The MAC address which will be allowed to communicate with the module.
Description	An optional description for the Device with the matching MAC address.

Table 6.7 – Security Configuration – MAC Access Control List

The MAC addresses can be entered manually, or the user can right-click and select the ***Browse*** option.

This will open the ***Network Interface Browser*** displaying all the network interfaces and their corresponding MAC addresses found on the local pc. The user can then select which MAC addresses to be added to the MAC Access Control List.

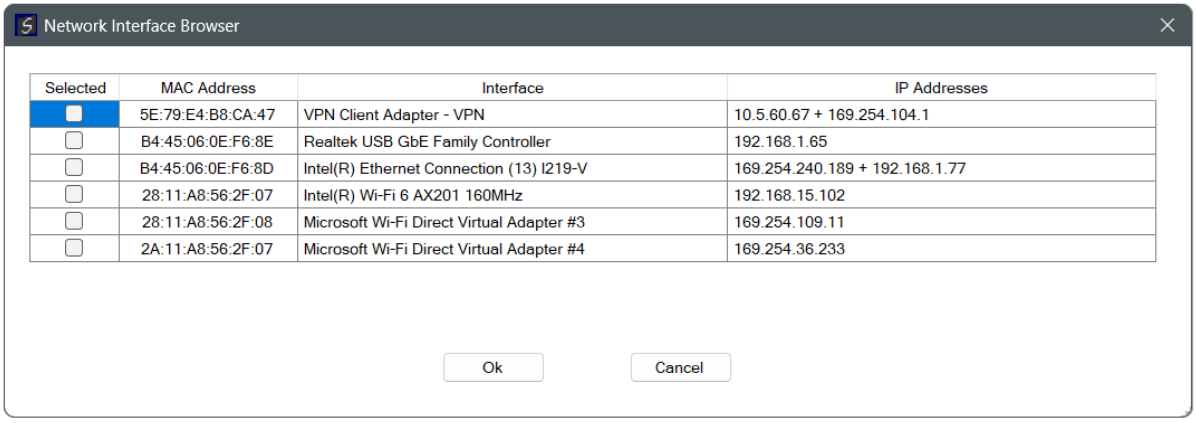


Figure 6.10 – Security Configuration – Network Interface Browser

6.1.2. Download Security Configuration

Once the security configuration has been completed it must be downloaded to the module.

When online with the module, the current state of the security will be indicated by the icon of the Security node in the project tree as follows:

Icon	Description
	Module Offline – Master Security Disabled
	Module Offline – Master Security Enabled
	Module Online – No Security Configuration Downloaded
	Module Online – Master Security Disabled
	Module Online – Master Security Enabled, User Authentication Uninitialized
	Module Online - Secured

Table 6.8 – Security Status Icons

To download the security configuration right-click on the Security item in the project tree and select the ***Download Security Configuration*** item.

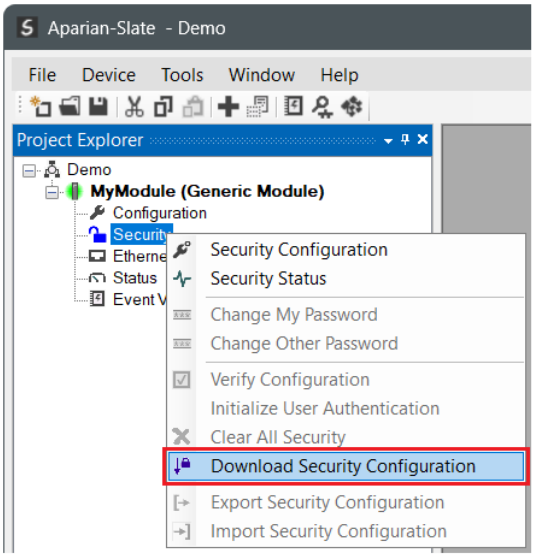


Figure 6.11 – Download Security Configuration

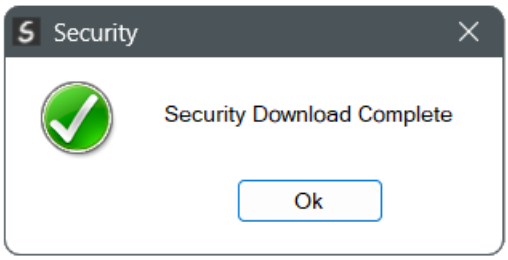


Figure 6.12 – Download Security Configuration Complete

6.1.3. Initialize User Authentication

The final step in securing the module is to initialize user authentication. This involves creating an Administrator password as well as setting the Default password to be used by all other users for their initial login.

To commence with this step, right-click on the Security node in the project tree and select the ***Initialize User Authentication*** option.

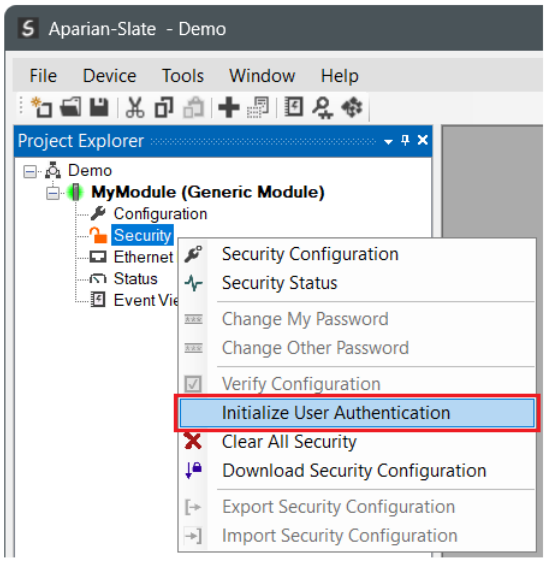


Figure 6.13 – Initialize User Authentication

The user (now deemed to be the Administrator) will need to configure a password for the fixed ***Administrator*** user.

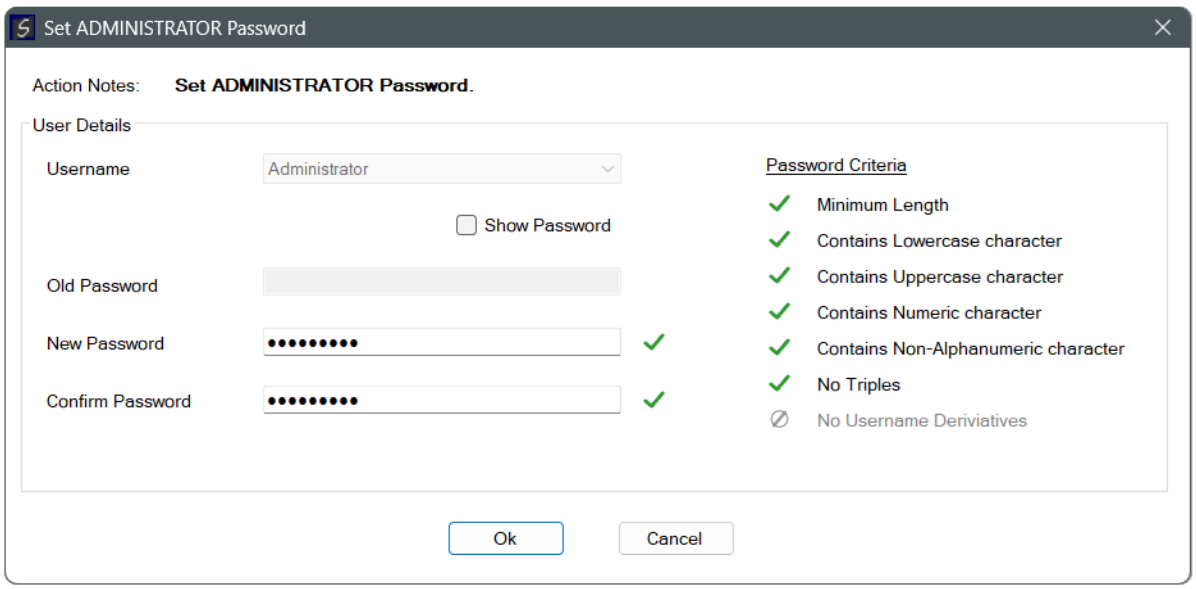


Figure 6.14 – Set Administrator Password.

As the new password is entered, it will be checked against the configured password rules. The icons adjacent to each password rule will indicate if the criteria are fulfilled or not, or not applicable, as shown below:

Icon	Description
	Criteria Not Applicable
	Failed Criteria
	Passed Criteria

Table 6.9 – Password Rule Status Icons.

A green tick icon will be displayed to the right of the ***New Password*** once it meets all the password criteria. A green tick adjacent to the ***Confirm Password*** will indicate that it matches the ***New Password***.

The entered passwords can either be displayed or masked by clicking on the ***Show Password*** check box.

Once the Administrator password has been set, then the ***Default*** password must be set. The ***Default*** password is the password to be used by all other users for their initial login.

The ***Default*** password is configured in similar manner and must also comply with the configured password rules.

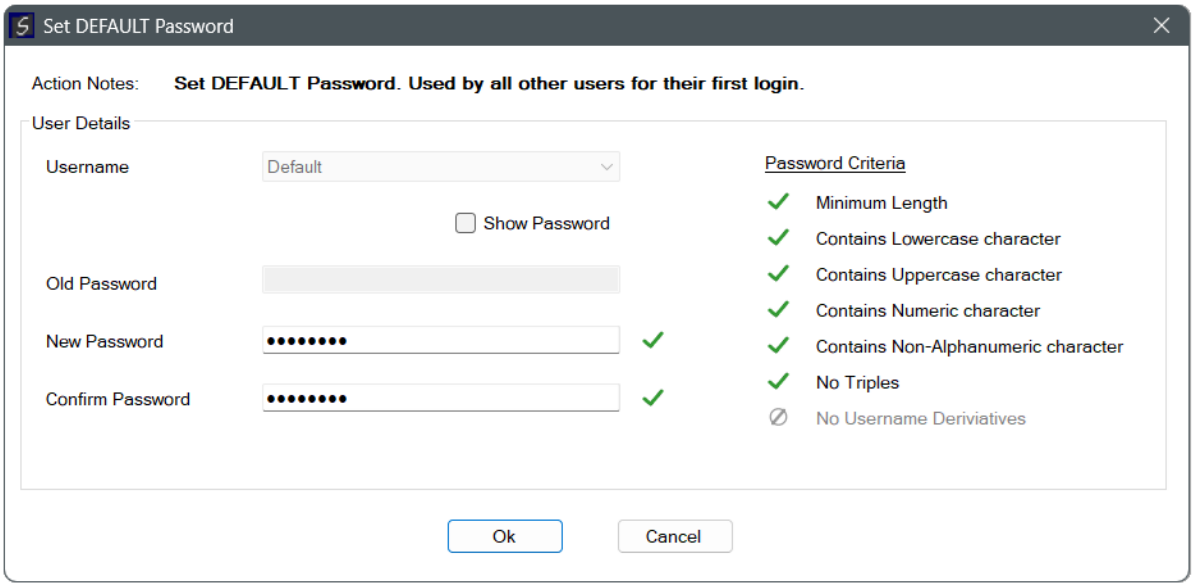


Figure 6.15 – Set Default Password.



NOTE: If required, the Administrator can change the default password individually for each user, by using the ***Change Other Password*** option.

On completion of the User Authentication Initialization step, the module will be secured indicated by the green Locked icon.

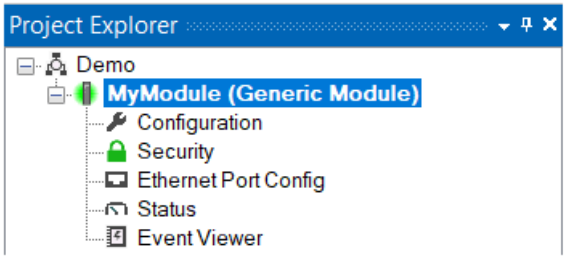


Figure 6.16 – Secured Module

6.2. OPERATION

To interact with a secured, and user initialized module, each user will need to first login.

6.2.1. INITIAL USER LOGIN

Similar, to an unsecured module, a user would open the existing Slate project, right-click on the module, and select ***Go Online***.

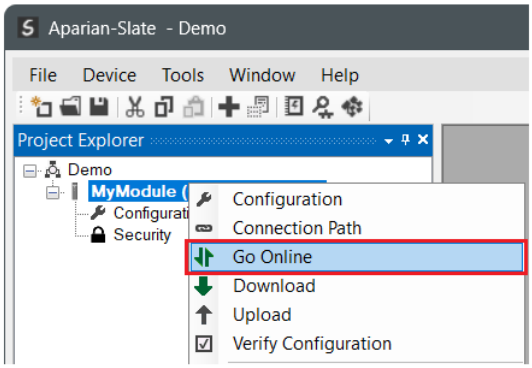


Figure 6.17 – Go Online

Since the module has been secured, it will then prompt the user to login.

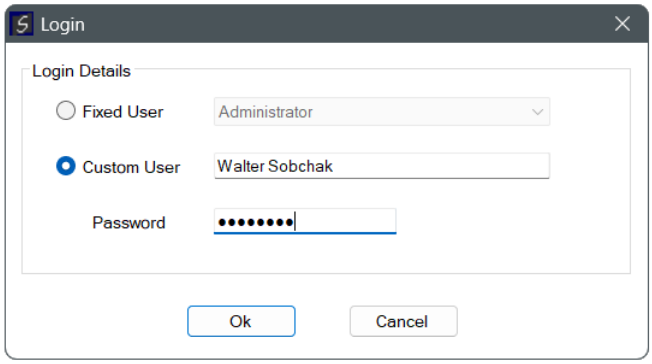


Figure 6.18 – Custom User Login

The user will need to select either one of the ***Fixed Users*** or select the ***Custom User*** option and then enter the Custom Username.

The user will then need to enter their password.

Should this be the initial login with the Default password, or if the Administrator has reset the password and selected the option for the user to change the password, then the Set Password form will open. The user will then need to set an appropriate password before continuing.

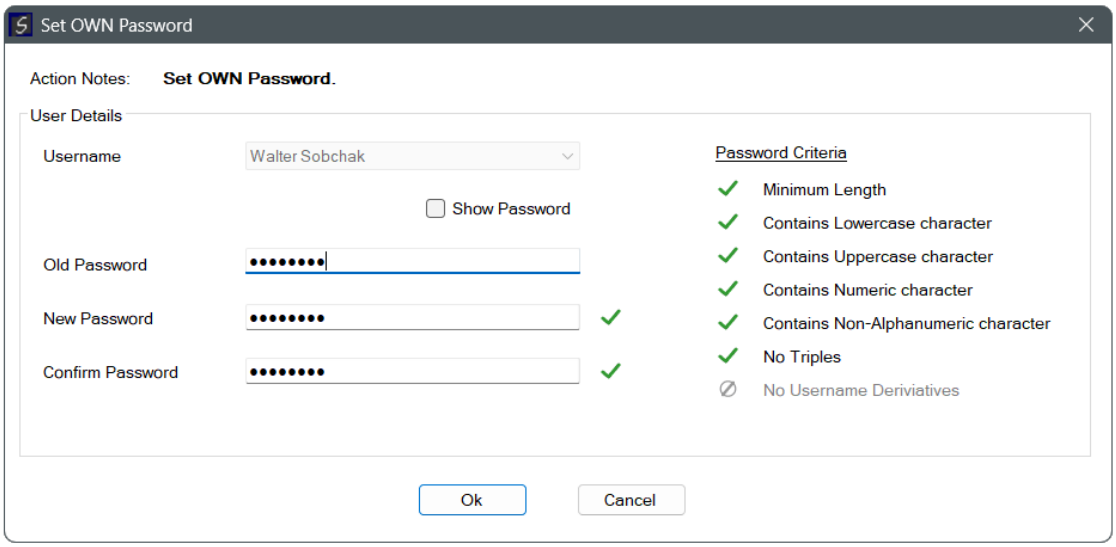


Figure 6.19 – Change Own Password

NOTE: The *Old Password* that needs to be entered on the Initial User Login, will be the same password that was used to login.

All users are permitted to change their password at any time by right-clicking on the ***Security*** node in the project tree and selecting the ***Change My Password*** option.

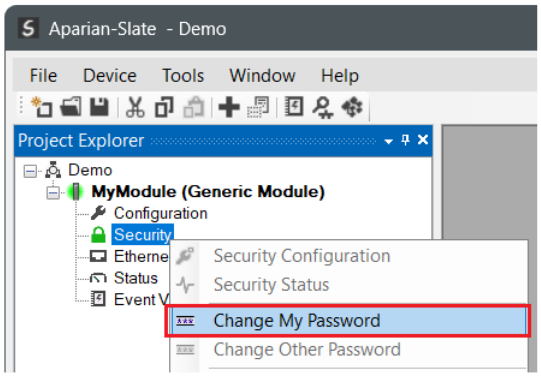


Figure 6.20 – Select Change My Password

6.2.2. ROLE SPECIFIC OPERATIONS

Users with Administrator, Engineer and Auditor roles are permitted to access additional functions not available by other users.

6.2.2.1. SECURITY STATUS

The Security Status can be viewed by right-clicking on the ***Security*** node in the project tree and selecting the ***Security Status*** option.

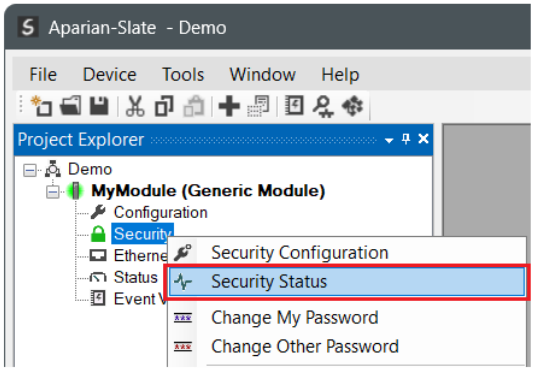


Figure 6.21 – Select Security Status

The Security Status window will open displaying various status and statistics regarding the security services.

The ***General*** tab displays an overview of the security status.

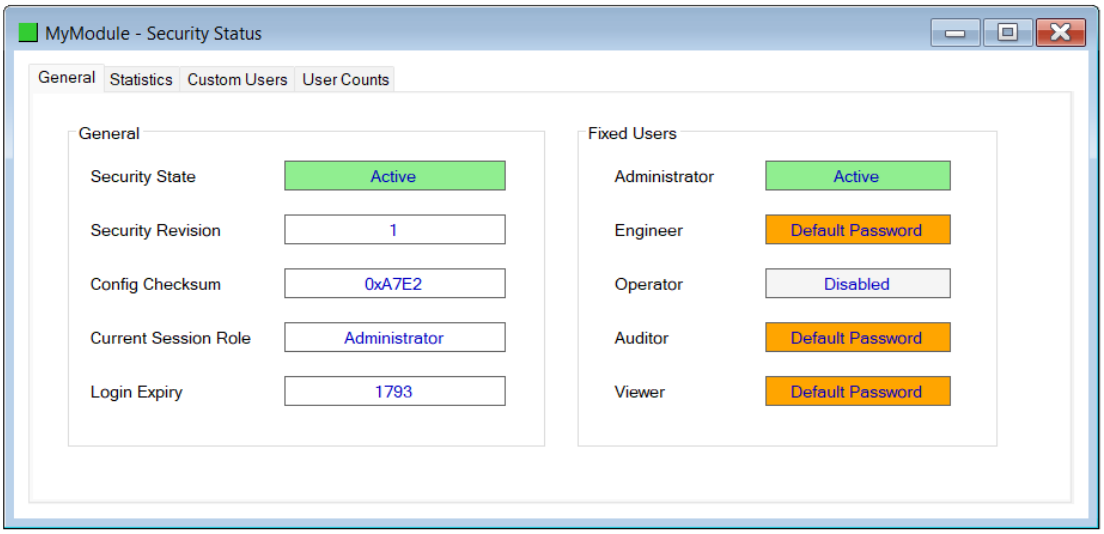


Figure 6.22 – Security Status – General

The General status tab displays the following:

Parameter	Description
General	
Security State	The current status of the security services: ***No Config*** No security configuration has been downloaded to the module. ***Disabled*** Security configuration has been downloaded to the module, but the master Enable Security option in the configuration has been disabled. ***Uninitialized*** The security configuration has been downloaded and enabled, but the Initialize User Authentication step has not yet been completed. ***Active*** The module has been successfully secured.
Security Revision	The revision of the security services currently running on the module. This is a function of the module's firmware revision.
Config Checksum	The checksum of the current security configuration.
Current Session Role	The role assigned to the user that is currently logged in.
Login Expiry	The time (seconds) until the user is automatically logged out. Only applicable if the ***Login Expire*** option has been enabled in the security configuration.

Fixed Users	
Fixed Users Status Administrator Engineer Operator Auditor Viewer	The status of each fixed user: ***Disabled*** The user has not been enabled in the security configuration. ***Default Password*** The user has not changed from their Default assigned password, or from when an Administrator reset their password. ***No Password*** No password exists for the user. This would typically occur when a fixed user is enabled after ***the User Authentication Initialization*** step has taken place. An Administrator would need to reset the password for this specific user. ***Active*** The user is enabled and has a valid password configured.

Table 6.10 – Security Status – General

The Statistics tab displays a number of security service statistics.

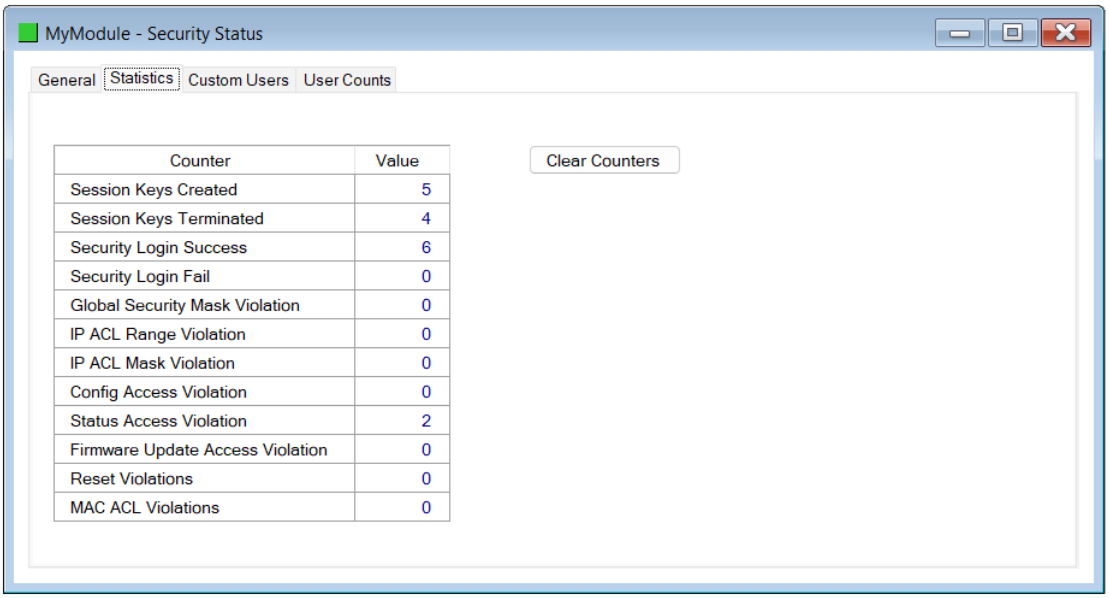


Figure 6.23 – Security Status – Statistics

NOTE: Only an Administrator is permitted to select the ***Clear Counters*** option.

Parameter	Description
Session Keys Created	The number of security session keys created between the configuration software and the module. Each time a user connects to the module with the configuration software, and the module has been secured, a new session key will be created.
Session Keys Terminated	The number of times when either the connection from the configuration software has been terminated or there was an error in the connection decryption. In both these cases the session keys will be deleted.
Security Login Success	The number of successful logins.
Security Login Fail	The number of failed login attempts.
Global Security Mask Violation	The number of times a connection request is received from another device on the Ethernet network, which has not been enabled in the module's global security mask configuration.
IP ACL Range Violation	Each time a request or packet is received from a device that has an IP address outside the IP ACL range, this number will be incremented.
IP ACL Mask Violation	Each time a request or packet is received from a device that has an IP address outside the Subnet ACL range, this number will be incremented.
Config Access Violation	The number of times a connection with the incorrect role privileges tried to access the module security configuration.
Status Access Violation	The number of times a connection with the incorrect role privileges tried to access the module security status.
Firmware Update Access Violation	The number of times an attempt to flash upgrade the module in an incorrect state (e.g., while running a class 1 EtherNet/IP connection) was made.
Reset Violations	The number of times an attempt to reset the module in an incorrect state (e.g., while running a class 1 EtherNet/IP connection) was made.
MAC ACL Violations	Each time a request or packet is received from a device that does not have a matching MAC address to those in the MAC ACL list, this number will increase.

Table 6.11 – Security Status – Statistics

The Custom Users status tab displays the following:

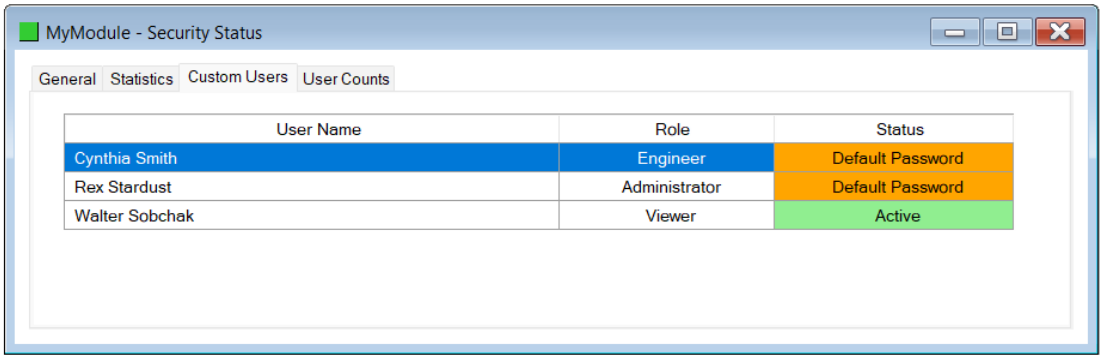


Figure 6.24 – Security Status – Custom Users

Parameter	Description
User Name	The configured user name.
Role	The configured role for this specific user.
Status	The status of the specific user: ***Default Password*** The user has not changed from their Default assigned password, or from when an Administrator reset their password. ***No Password*** No password exists for the user. This would typically occur when a user is added after the User Authentication Initialization has taken place. An Administrator would need to reset the password for this specific user. ***Active*** The user is enabled and has a valid password configured.

Table 6.12 – Security Status – Custom Users

The User Counts status tab displays the following:

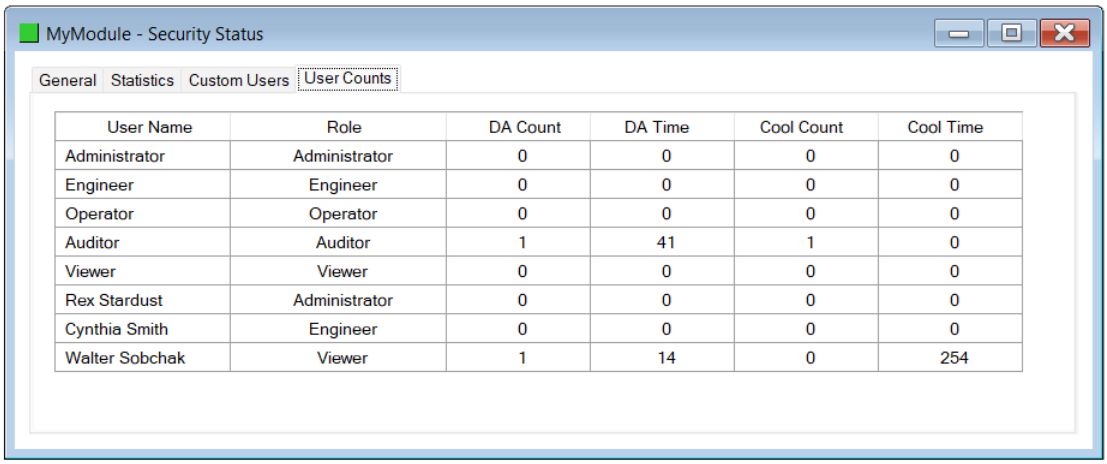


Figure 6.25 – Security Status – User Counts

The following statistics are displayed for each user:

Parameter	Description
User Name	The configured user name.
Role	The configured role for this specific user.
DA Count	The number of failed login occurrences with respect to Disabling the account.

DA Time	The time (seconds) since the last failed login occurrence with respect to Disabling the account.
Cool Count	The number of failed login occurrences with respect to account Cool Down period.
Cool Time	The time (seconds) remaining for the account cool down period.

Table 6.13 – Security Status – User Count

6.2.3. Change Other Password

An Administrator can change (or reset) any user's password by right-clicking on the Security node in the project tree and selecting the ***Change Other Password*** option.

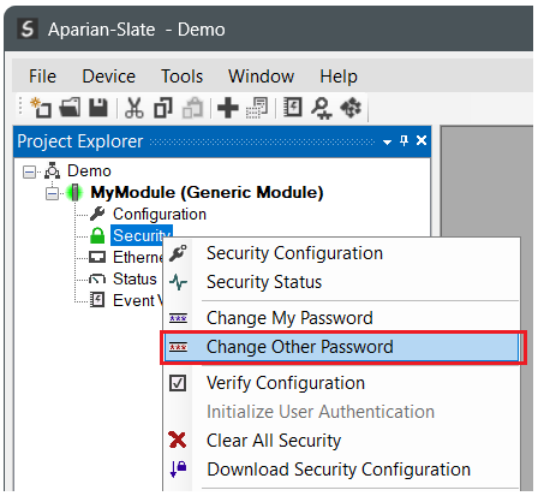


Figure 6.26 – Select Change Other Password.

The Set Password form will open, and the Administrator can then select the User's password to change. The Administrator will need to re-enter their password as well as the new password for the user.

The ***Require User to Change Password*** option forces that user to change the password the next time they login.

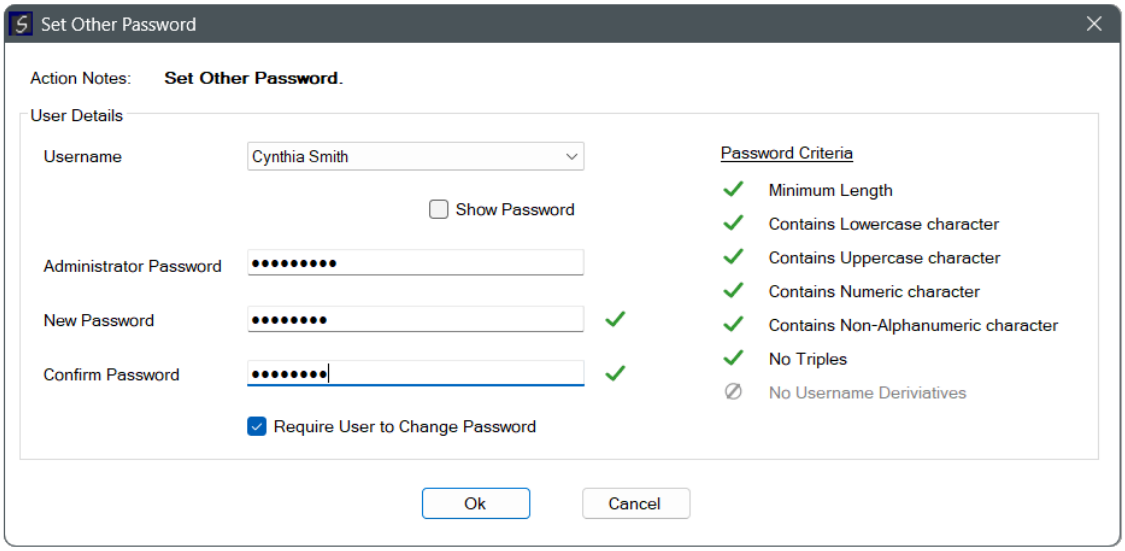


Figure 6.27 – Set Other Password

6.2.4. Clear All Security

An Administrator can remove all the security from a module and return it to an unsecured state. This is done by right-clicking on the Security node in the project tree and selecting the ***Clear All Security*** option.

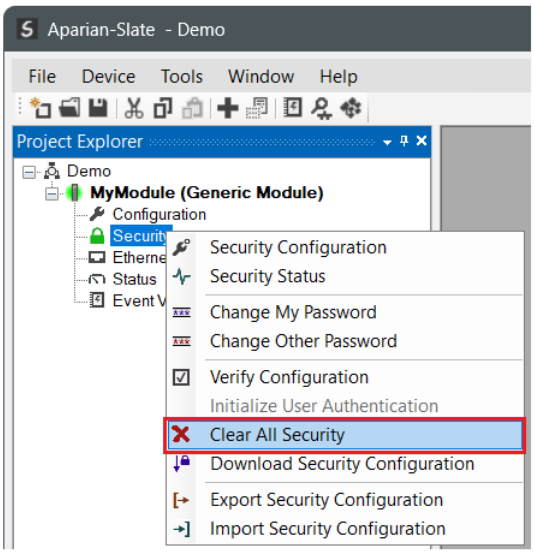


Figure 6.28 – Select Clear All Security.

This action will require the Administrator to enter their password. The security configuration and all passwords will be removed with this action.

NOTE: This process will remove all module security configuration including all passwords.

6.3. EVENT LOG

All security related events are added to the module's event log.

NOTE: Only an Administrator is permitted to select the ***Clear*** the event log.

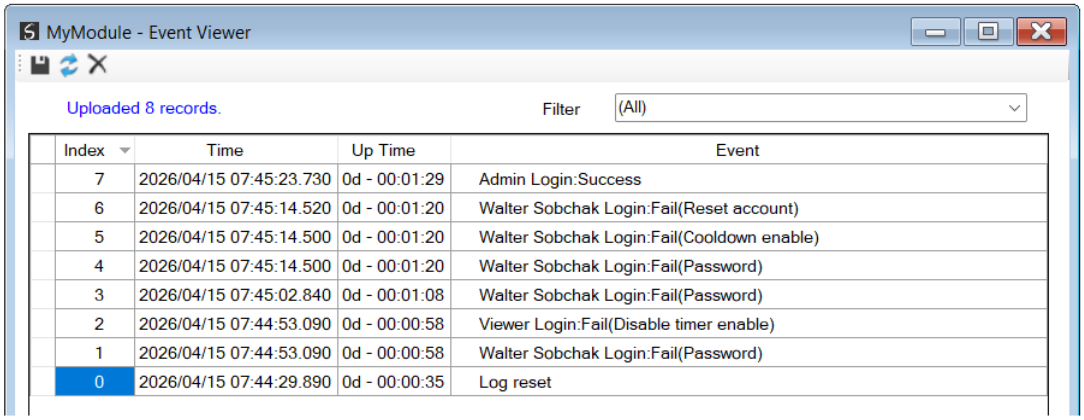


Figure 6.29 – Event Log

6.4. RESET SECURITY CONFIGURATION

The module's security configuration can be reset (cleared) by powering-up the module with DIP switches 2, 3, and 4 set to the ON position.

NOTE: This process will remove all module application configuration and security configuration including all passwords.

NOTE: Once the module has booted-up, the DIP switches should be returned to normal (all OFF). New security configuration cannot be downloaded with the Reset DIP switch combination active.

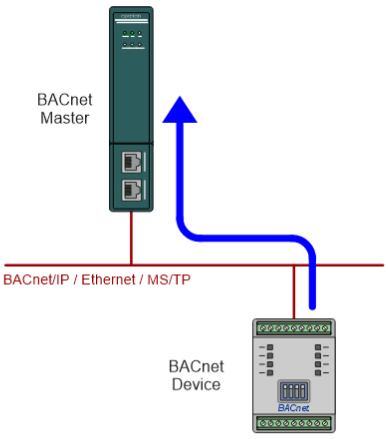
7. OPERATION

7.1. BACNET

The BACnet Master is able to exchange data with BACnet devices by reading and writing object properties.

7.1.1. INPUT DATA

When an object property value is received from a BACnet device it is considered to be **Input** data. This data can then be sent to the BACnet Master's primary interface, either an EtherNet/IP or Modbus device.



	En	Function	Scan	Object	Start	Count	Property	Format	COV Lifetime	COV Inc.	I	Is	O	Os
1	☑	Read	A	Analog Input	1	2	Present Value	w/Status			0	16	0	0
2	☑	Write Change		Binary Output	1	4	Present Value	Standard			16	0	0	16

Figure 7.1 – BACnet Input data

Receiving property data from a BACnet device can be configured in the device's Property Configuration using a number of functions:

7.1.1.1. READ

The specified object property is explicitly read from the BACnet device at the configured ***Scan*** rate, and the received data is mapped into the corresponding BACnet Master's input data.

7.1.1.2. COV (CHANGE OF STATE SUBSCRIPTION)

The BACnet master subscribes to the specified device object's COV (Change of State) service. Each time a property of that object changes, the BACnet device will send the property value to the BACnet Master in an unsolicited manner.

Should the transmitted property match the configured property, then the received data is mapped into the corresponding BACnet Master's input data.

The configured COV Lifetime parameter is the time (in seconds) that the subscription will last. At the expiration of this time, the BACnet Master will automatically re-subscribe to the service.

Property Map Configuration

	En	Function	Scan	Object	Start	Count	Property	Format	COV Lifetime	COV Inc.	I	Is	O	Os
1	☑	CoV		Analog Input	1	1	Present Value	w/Status	60		0	8	0	0
▸*	☐													

Figure 7.2 – COV Function



NOTE: Not all BACnet devices support the COV subscription service.

7.1.1.3. COV PROP (CHANGE OF STATE PROPERTY SUBSCRIPTION)

The BACnet master subscribes to the specified device object property COV (Change of State) service. Each time that specific property of that object changes, the BACnet device will send the property value to the BACnet Master in an unsolicited manner. The received data is mapped into the corresponding BACnet Master's input data.

The configured COV Lifetime parameter is the time (in seconds) that the subscription will last. At the expiration of this time, the BACnet Master will automatically re-subscribe to the service.



NOTE: The ***COV Prop*** function differs only from the ***COV*** function in that it subscribes only to the specific property of the object, rather than all properties of that object.

Property Map Configuration

	En	Function	Scan	Object	Start	Count	Property	Format	COV Lifetime	COV Inc.	I	Is	O	Os
1	☑	CoV Prop		Analog Input	1	1	Present Value	w/Status	60		0	8	0	0
▸*	☐													

Figure 7.3 – COV Prop. Function

NOTE: Not all BACnet devices support the COV Property subscription service.

7.1.1.4. COV Prop Inc (Change of State Property Subscription with Increment)

The BACnet master subscribes to the specified device object property COV (Change of State) service and specifies the required Increment (deadband) value. Each time that specific property of that object changes by at least the specified Increment value, the BACnet device will send the property value to the BACnet Master in an unsolicited manner. The received data is mapped into the corresponding BACnet Master's input data.

The configured COV Lifetime parameter is the time (in seconds) that the subscription will last. At the expiration of this time, the BACnet Master will automatically re-subscribe to the service.

NOTE: The ***COV Prop Inc*** function differs only from the ***COV Prop*** function in that it also specifies the required increment (deadband) value that the property must change by, before the property value is transmitted.

Property Map Configuration

	En	Function	Scan	Object	Start	Count	Property	Format	COV Lifetime	COV Inc.	I	Is	O	Os
1	☑	CoV Prop.Inc		Analog Input	1	1	Present Value	w/Status	60	3	0	8	0	0
▸*	☐													

Figure 7.4 – COV Prop. Inc. Function

NOTE: Not all BACnet devices support the COV Property subscription service with an increment specification.

7.1.1.5. MAP

The Map function does not initiate any BACnet action and is used only to map received object property data, into the corresponding BACnet Master's input data

The received data may be as a result of another function, for example reading a Group.

In the example below, Group 1 is Read at an interval defined by Scan class A. The BACnet device has Group 1 pre-configured (of fixed) to contain 2 Analog inputs (1-2) and 4 Binary Inputs (1-4). The BACnet device will then send the property data for all these objects in response to the Group 1 read.

The BACnet master will use the Map functions to determine where this data is to be mapped in its input data.

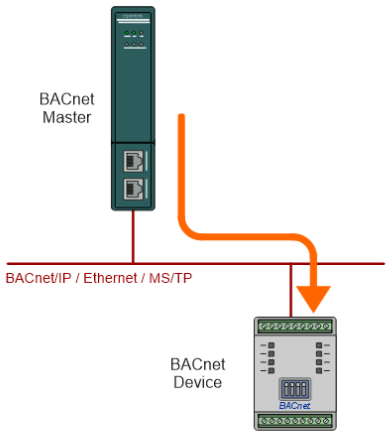
Property Map Configuration

	En	Function	Scan	Object	Start	Count	Property	Format	COV Lifetime	COV Inc.	I	Is	O	Os
1	☑	Read	A	Group	1	1	Present Value	w/Status			0	4	0	0
2	☑	Map		Analog Input	1	2	Present Value	w/Status			4	16	0	0
3	☑	Map		Binary Input	1	4	Present Value	w/Status			20	32	0	0
▸*	☐													

Figure 7.5 – Map function

7.1.2. Output Data

When an object property value is written to a BACnet device it is considered to be **Output** data. This data may originate from the BACnet Master's primary interface, either an EtherNet/IP or Modbus device.



	En	Function	Scan	Object	Start	Count	Property	Format	COV Lifetime	COV Inc.	I	Is	O	Os
1	☑	Read	A	Analog Input	1	2	Present Value	w/Status			0	16	0	0
2	☑	Write Change		Binary Output	1	4	Present Value	Standard			16	0	0	16

Figure 7.6 – BACnet Output data

7.1.2.1. WRITE

The specified object property value is explicitly written to the BACnet device at the configured ***Scan*** rate. This data is contained within the corresponding BACnet Master's output data.

Property Map Configuration

	En	Function	Scan	Object	Start	Count	Property	Format	COV Lifetime	COV Inc.	I	Is	O	Os
1	☑	Write	A	Analog Output	1	1	Present Value	w/Status			0	4	0	4
▸*	☐													

Figure 7.7 – Write function

7.1.2.2. WRITE CHANGE

The specified object property value is explicitly written to the BACnet device only when the source data changes. This data is contained within the corresponding BACnet Master's output data.

Property Map Configuration

	En	Function	Scan	Object	Start	Count	Property	Format	COV Lifetime	COV Inc.	I	Is	O	Os
1	☑	Write Change		Analog Output	1	1	Present Value	w/Status			0	4	0	4
▸*	☐													

Figure 7.8 – Write Change function

7.1.3. FORMAT

The format field for each function specifies how the data formatted and/or mapped.

Property Map Configuration

	En	Function	Scan	Object	Start	Count	Property	Format	COV Lifetime	COV Inc.	I	Is	O	Os
1	☑	Read	A	Analog Value	1	1	Present Value	Standard			0	4	0	0
2	☑	Read	A	Analog Input	1	1	Present Value	w/Status			4	8	0	0
3	☑	Read	A	Binary Input	1	12	Present Value	Packed			12	4	0	0
4	☑	Write Change		Analog Output	1	1	Present Value	w/Status			16	4	0	4

Figure 7.9 – Format

7.1.3.1. STANDARD

When the Standard format option is selected, only the input or output data is transferred. The actual format of the data depends on the object. For example, Analog Input will be a REAL value, and a Binary Output will be a BOOLEAN value. No status information is provided by the BACnet Master.

7.1.3.2. W/STATUS (STANDARD WITH STATUS)

The w/Status (Standard Data with Status) format transfers the input or output data according to the specified object, but also provides Status information about the transaction. This Status information always resides in the Input data section irrespective of whether the action makes use of input data (e.g. Read) or output data (e.g. Write).

The Property Status information is 4 bytes in length, and is described in the Data Format section of this chapter.

7.1.3.3. PACKED

The ***Packed*** format option is available only for Binary objects and is used to make the Input and Output assembly data more efficient. With the Standard and Standard w/Status options, each property is padded so as to start at a 32-bit boundary. So binary value values, although BOOLEAN in nature will consume 4 bytes each.

In the Packed format, contiguous Binary objects are packed together in a BOOLEAN array. The first object will occupy bit 0, the next bit 1 and so on.

For example, a Read function on Binary Inputs 1-12, will occupy the first 12 bits of a BOOLEAN array, and will consume 4 bytes in total (1.5 bytes rounded up to a 32-bit boundary). Whereas, with the Standard format, this would consume 48 bytes.

NOTE: When packed it selected, there will be no status mapped to provide information about the specific property data exchange.

7.2. ETHERNET/IP TARGET

A controller (e.g. Logix controller) can own the BACnet Master over EtherNet/IP using up to 25 Class 1 EtherNet/IP connections when the BACnet Master is operating as an EtherNet/IP target. This will allow the BACnet Master to exchange data with the controller using the input and output assembly of the Class 1 EtherNet/IP connections.

Input data, typically read from a BACnet device, appears in the EtherNet/IP controller's Input assembly.

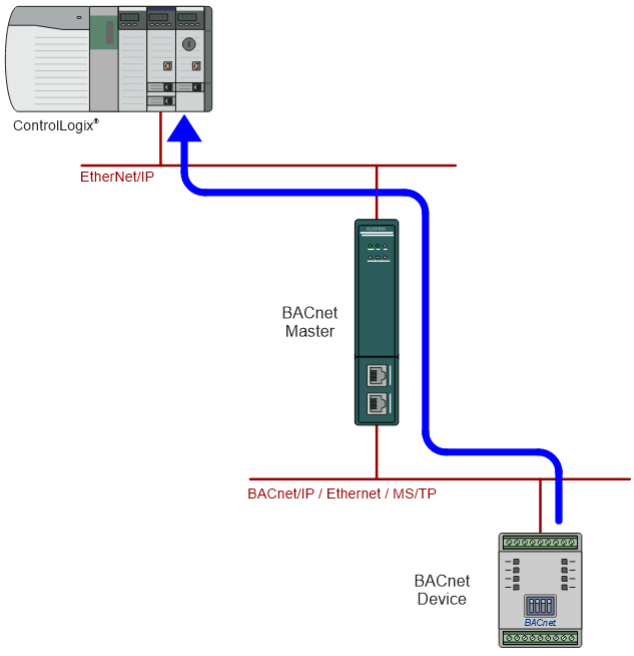


Figure 7.10 – BACnet Input data – Master as EtherNet/IP Target

Data received from the EtherNet/IP controller's output assembly can be written to a BACnet device.

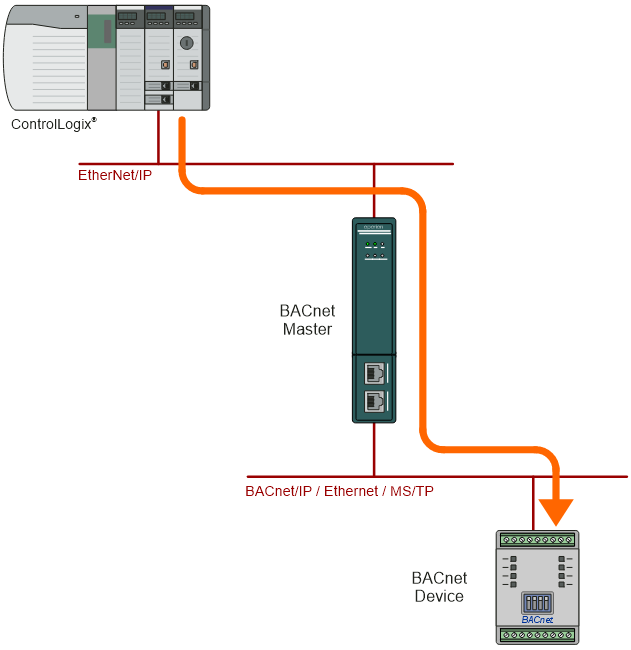


Figure 7.11 – BACnet Output data – Master as EtherNet/IP Target

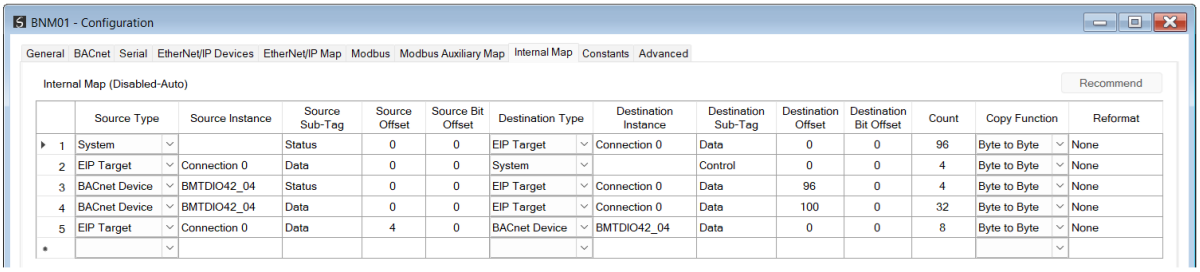
NOTE: When using EtherNet/IP Target, it is recommended to select the ***Enable Auto Internal Mapping*** option in the BACnet Master's Advance configuration tab, or to use the ***Recommend*** button in the Internal Map configuration. This will automatically map and reformat all the required data in the Internal Map.

7.2.1. Class 1 Assembly Mapping

When the module operates in a Logix "owned" mode the Logix controller will establish a class 1 cyclic communication connection to the BACnet Master. Up to 25 input and output assemblies are exchanged at a fixed interval (RPI).

NOTE: The module input and output assembly of each connection will be an undecorated array of bytes. The imported Logix routine (generated by Slate) will copy this data to the input and output assemblies.

Once the generate L5X file has been imported (which will match the Internal Mapping in the configuration), the user will be able to use the tags generated for the specific BACnet Master.



	Source Type	Source Instance	Source Sub-Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Sub-Tag	Destination Offset	Destination Bit Offset	Count	Copy Function	Reformat
1	System		Status	0	0	EIP Target	Connection 0	Data	0	0	96	Byte to Byte	None
2	EIP Target	Connection 0	Data	0	0	System		Control	0	0	4	Byte to Byte	None
3	BACnet Device	BMTDIO42_04	Status	0	0	EIP Target	Connection 0	Data	96	0	4	Byte to Byte	None
4	BACnet Device	BMTDIO42_04	Data	0	0	EIP Target	Connection 0	Data	100	0	32	Byte to Byte	None
5	EIP Target	Connection 0	Data	4	0	BACnet Device	BMTDIO42_04	Data	0	0	8	Byte to Byte	None
*													

Figure 7.12 – Internal Map – EtherNet/IP Target

For each BACnet device, a dedicated tag will be created including the Device Status, Input Data (including Property Statuses where applicable) and Output Data.

An example Logix tag is shown below which was generated from a device with the following property Configuration:

En	Function	Scan	Object	Start	Count	Property	Format
☑	Read	A	Analog Value	1	3	Present Value	w/Status
☑	Write Change		Analog Output	1	2	Present Value	w/Status

	Name	Value	Force Mask	Style	Data Type
Device Status	BNM01_BMTDIO42_04	{...}	{...}		APBNM01Dev403D219B
	BNM01_BMTDIO42_04.Status	{...}	{...}		APBACnetDeviceStatus
	BNM01_BMTDIO42_04.Status.Online	0		Decimal	BOOL
	BNM01_BMTDIO42_04.Status.DataExchange	0		Decimal	BOOL
	BNM01_BMTDIO42_04.Status.PropertyUpdates	0		Decimal	INT
Input	BNM01_BMTDIO42_04.In	{...}	{...}		APBNM01Dev403D219BIn
	BNM01_BMTDIO42_04.In.AV001PVStatus	{...}	{...}		APBACnetPropertyStatus
	BNM01_BMTDIO42_04.In.AV001PVStatus.Success	0		Decimal	BOOL
	BNM01_BMTDIO42_04.In.AV001PVStatus.Fail	0		Decimal	BOOL
	BNM01_BMTDIO42_04.In.AV001PVStatus.COVNotification	0		Decimal	BOOL
	BNM01_BMTDIO42_04.In.AV001PVStatus.SubscriptionActive	0		Decimal	BOOL
	BNM01_BMTDIO42_04.In.AV001PVStatus.DataStale	0		Decimal	BOOL
	BNM01_BMTDIO42_04.In.AV001PVStatus.SuccessCount	0		Decimal	INT
	BNM01_BMTDIO42_04.In.AV001PV	0.0		Float	REAL
	BNM01_BMTDIO42_04.In.AV002PVStatus	{...}	{...}		APBACnetPropertyStatus
	BNM01_BMTDIO42_04.In.AV002PV	0.0		Float	REAL
	BNM01_BMTDIO42_04.In.AV003PVStatus	{...}	{...}		APBACnetPropertyStatus
	BNM01_BMTDIO42_04.In.AV003PV	0.0		Float	REAL
	BNM01_BMTDIO42_04.In.AO001PVWStatus	{...}	{...}		APBACnetPropertyStatus
	BNM01_BMTDIO42_04.In.AO002PVWStatus	{...}	{...}		APBACnetPropertyStatus
Output	BNM01_BMTDIO42_04.Out	{...}	{...}		APBNM01Dev403D219BOut
	BNM01_BMTDIO42_04.Out.AO001PV	0.0		Float	REAL
	BNM01_BMTDIO42_04.Out.AO002PV	0.0		Float	REAL

Figure 7.13 – Logix – BACnet Device Tag

7.3. ETHERNET/IP ORIGINATOR

The BACnet Master module can operate as an EtherNet/IP originator. In this mode the module can exchange data from the BACnet network with EtherNet/IP devices using either the input and output assemblies of the Class 1 EtherNet/IP connection to the device or using explicit (Class 3 or UCMM) EtherNet/IP messages.

7.3.1. ETHERNET/IP CLASS 1 CONNECTIONS

In the example below, the BACnet Master is exchanging BACnet device data with owned EtherNet/IP I/O devices.

Once the EtherNet/IP Class 1 connections are configured and established, then any Internal Map copying to an EtherNet/IP device will be written to the output assembly of the selected EtherNet/IP device (Originator to Target). In the Internal Map the user will specify which device instance is the target and at what offset the data must be written in the output assembly.

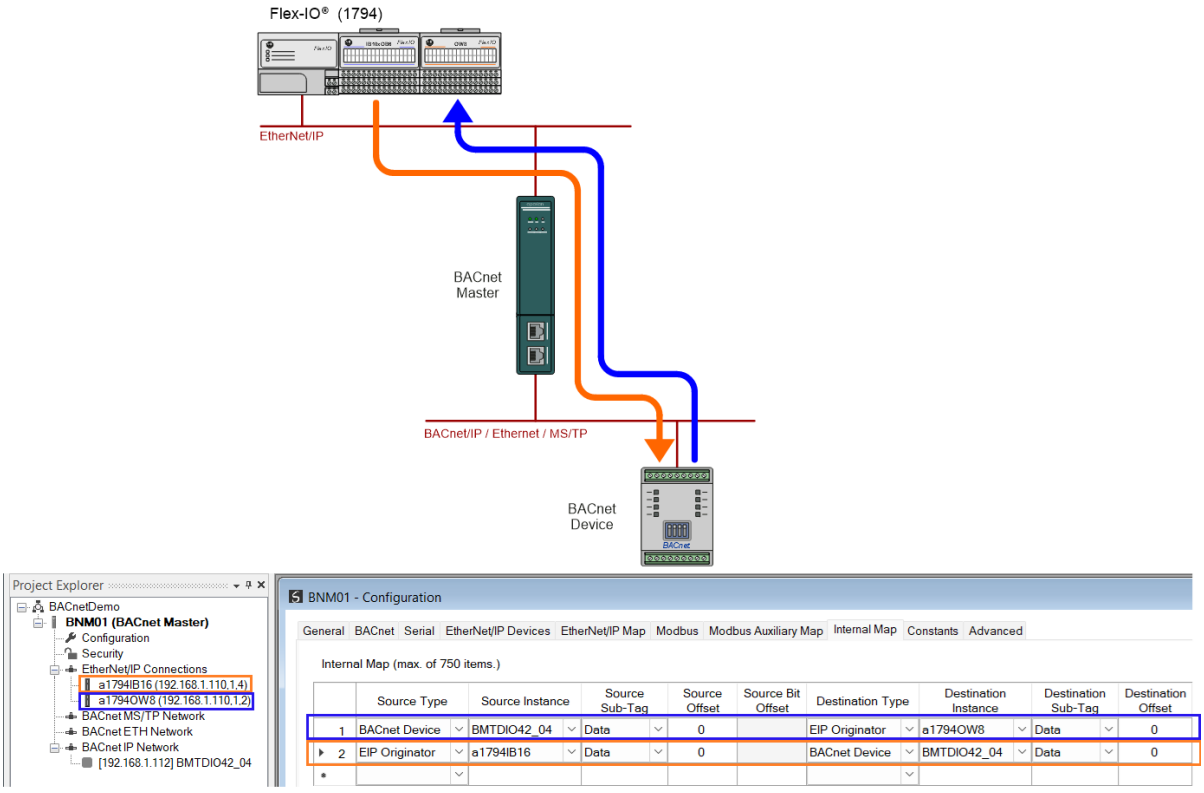


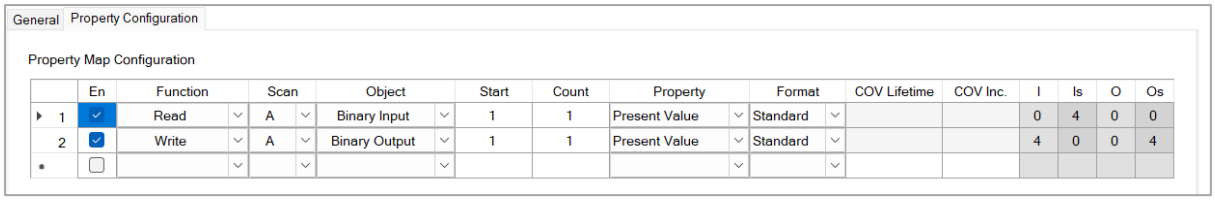
Figure 7.14 – Internal Mapping - EtherNet/IP Originator

7.3.2. EXPLICIT ETHERNET/IP MESSAGING

When using the EtherNet/IP Explicit Messaging, the user can configure up to 10 EtherNet/IP devices which will be used for the Explicit Messaging. This configuration is accessed in the *EtherNet/IP Devices* tab. Following this, the EtherNet/IP Map of explicit messages needs to be configured. The Explicit Messaging uses the internal data space (IDS) where data can be stored for exchanges between the explicit EtherNet/IP devices and the BACnet devices.

The Input and Output IDS Offset is where the Explicit EtherNet/IP device data will be read from, or written to. The data in the IDS can then, in turn, be copied to or from the BACnet device(s) using the Internal Mapping to copy the BACnet device data.

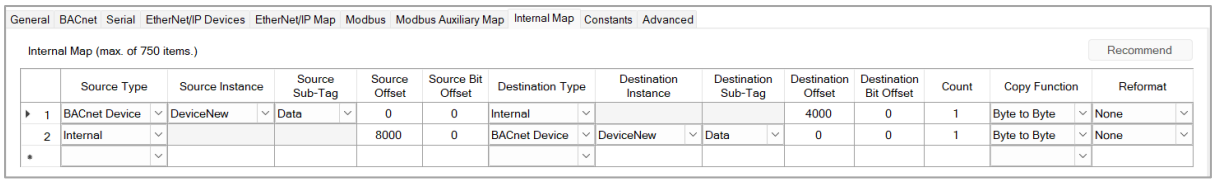
In the example below, a BACnet device has been configured to read one Binary Input and write one Binary Output.



	En	Function	Scan	Object	Start	Count	Property	Format	COV Lifetime	COV Inc.	I	Is	O	Os
1	☑	Read	A	Binary Input	1	1	Present Value	Standard			0	4	0	0
2	☑	Write	A	Binary Output	1	1	Present Value	Standard			4	0	0	4
*	☐													

Figure 7.15 – BACnet device data exchange

The Binary Input data has been mapped to Internal Data Space (IDS) register 4000 and Binary Output data has been mapped from IDS register 8000.



	Source Type	Source Instance	Source Sub-Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Sub-Tag	Destination Offset	Destination Bit Offset	Count	Copy Function	Reformat
1	BACnet Device	DeviceNew	Data	0	0	Internal			4000	0	1	Byte to Byte	None
2	Internal			8000	0	BACnet Device	DeviceNew	Data	0	0	1	Byte to Byte	None
*													

Figure 7.16 – Internal Mapping

In the EtherNet/IP Map, the data from IDS register 4000 (BACnet device Binary Input) is written to the Logix tag DI[0]. The data from Logix tag DO[0] is written to IDS register 8000 (BACnet device Binary Output).

General | BACnet | Serial | EtherNet/IP Devices | EtherNet/IP Map | Modbus | Modbus Auxiliary Map | Internal Map | Constants | Advanced

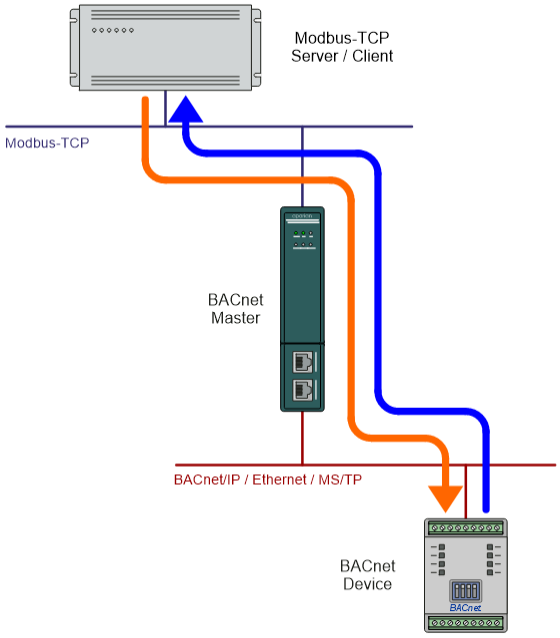
Explicit EtherNet/IP Map (max. of 50 items.)

	Device	Function	Scan	Service	Class	Instance	Attribute	Input Offset	Get Length	Output Offset	Set Length	Data Type	Tag / Static Value	...
1	MyL85	Read Tag	A					8000	1				DO[0]	...
2	MyL85	Write Tag	A							4000	1		DI[0]	...
3	TSM	Get	A		0x0001	1	3	2400	2					
▸*														

Figure 7.17 – EtherNet/IP device data exchange

7.4. MODBUS

When the BACnet Master has the Primary Interface set to either Modbus Client or Modbus Server, then the BACnet device data can be mapped to and from configurable internal Modbus Registers using the Internal Map.



General | BACnet | Serial | EtherNet/IP Devices | EtherNet/IP Map | Modbus | Modbus Auxiliary Map | Internal Map | Constants | Advanced

Internal Map (Disabled-Auto) — Recommend

	Source Type	Source Instance	Source Sub-Tag	Source Offset	Source Bit Offset	Destination Type	Destination Instance	Destination Sub-Tag	Destination Offset	Destination Bit Offset	Count	Copy Function	Reformat
▸ 1	System		Status	0	0	MB Register	HR	Data	1000	0	96	Byte to Byte	None
2	MB Register	HR	Data	1048	0	System		Control	0	0	4	Byte to Byte	None
3	BACnet Device	BMTDIO42_04	Status	0	0	MB Register	HR	Data	1050	0	4	Byte to Byte	None
4	BACnet Device	BMTDIO42_04	Data	0	0	MB Register	HR	Data	1052	0	32	Byte to Byte	None
5	MB Register	HR	Data	1068	0	BACnet Device	BMTDIO42_04	Data	0	0	8	Byte to Byte	None

Figure 7.18 – Modbus Client / Server

In the case of a Modbus Client, the Modbus Auxiliary map will also need to be configured to exchange this data with a remote Modbus Server.

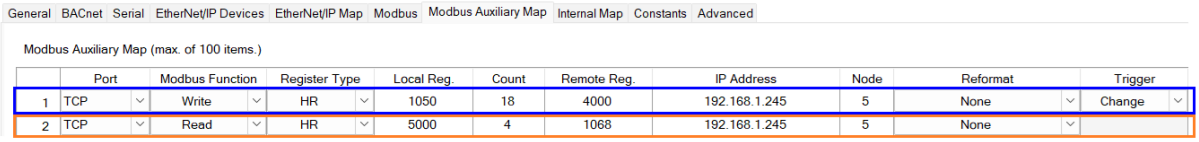


Figure 7.19 – Modbus Auxiliary Map

7.5. DATA FORMATS

The following tables describe the raw format of the data structures that can be mapped in the Internal Map table configuration.

7.5.1. SYSTEM STATUS

The System Status information can be mapped by setting the Source Type to ***System***.

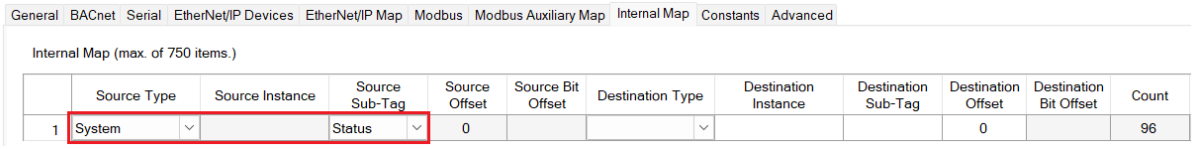


Figure 7.20 – System Status Information

The module's system status information has the following format.

Parameter	Data Type	Description
Status	DINT	Module Status. Bit 0 – Module Config Valid Bit 1 – EtherNet/IP Originator Comms Ok Bit 2 – Modbus Comms Ok Bit 3 – EtherNet/IP Target Comms Ok Bit 4 – Controller in Run Mode. Bit 5 – NTP Ok Bit 6 – BACnet Active Bit 7 – BACnet Running
BACnet Cycle Time	DINT	The BACnet Cycle Time in microseconds.
Frame Rx Count	DINT	The number of BACnet Frames received.
Frame Update Count	DINT	The number of Frame Updates (Input (Send) data changed)
Checksum Errors	DINT	The number of detected Frame Checksum errors.
Ident Frame Count	DINT	The number of BACnet Ident frames received.

Data Frame Count	DINT	The number of BACnet Data frames received.
Unknown Command Count	DINT	The number of unknown BACnet commands received.
Idle Timeout Count	DINT	The number of BACnet timeout occurrences.
Idle State Change Count	DINT	The number of INETRBUS Idle State changes.

Table 7.1 – System Status Format

7.5.2. System Control

The System Control information can be mapped by setting the Destination Type to ***System***.

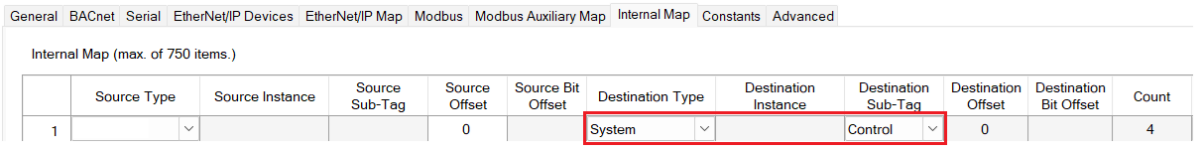


Figure 7.21 – System Control Information

The module's system control structure has the following format.

Parameter	Data Type	Description
Control Command	INT	Reserved
MappingCRC	INT	Mapping CRC checksum, which is used to ensure the mapped data structure (within say a Controller) matches that of the module. The action of the BACnet module when there is a mismatch is configured by the Action on Mapping Mismatch parameter. **NOTE:** Only relevant when the primary interface is set to EtherNet/IP Target.

Table 7.2 – System Control Format

7.5.3. Device Status

The BACnet Device Status information can be mapped by setting the Source Type to ***BACnet Device,*** selecting the required device, and setting the Source Sub-Tag to ***Status***.

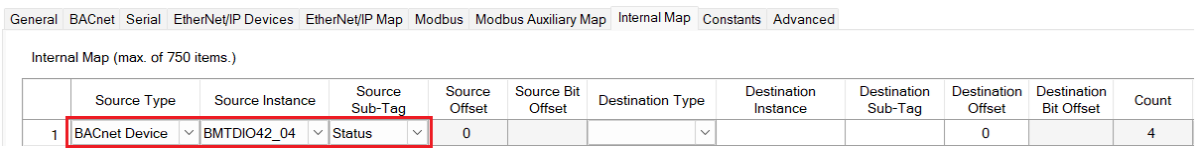


Figure 7.22 – Device Status Information

The BACnet Device Status structure has the following format.

Parameter	Data Type	Description
Status	INT	Device Status. Bit 0 – Online. The device is communicating on the network. Bit 1 – DataExchange. All configured object properties are being exchanged correctly.
Property Updates	INT	Increments each time a configured object property is updated.

Table 7.3 – Device Status Format

7.5.4. PROPERTY STATUS

The Property Status is automatically included for each property instance, if the property format is set to ***w/Status*** (Standard with Status). In the case of input data it precedes the property data. The Property Status structure is formatted as follows:

Parameter	Data Type	Description
Status Flags	INT	Status Flags Bit 0 – Success – Last transaction was successful. Bit 1 – Fail – Last transaction failed. Bit 2 – COV – Last update was via COV Notification. Bit 3 – COV Active – COV Subscription is currently Active. Bit 4 – Stale Data – This property has not been updated within the device's configured ***Data Stale Time*** parameter.
Update Count	INT	The number of successful property updates. This counter rolls-over at 65535.

Table 7.4 – Property Status Format

8. DIAGNOSTICS

8.1. LEDS

The module provides six LEDs for diagnostics purposes as shown in the front view figure below. A description of each LED is provided in the table below.

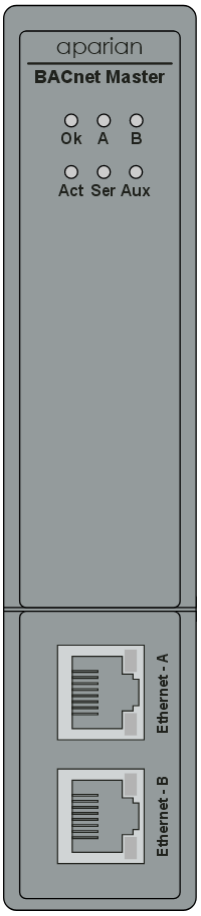


Figure 8.1 – BACnet Master front view

LED	Description
Ok	The module LED will provide information regarding the system-level operation of the module. If the LED is **red**, then the module is not operating correctly. For example, if the module application firmware has been corrupted or there is a hardware fault the module will have a red Module LED. If the LED is briefly **flashing red**, and then returning to either flashing green or solid green, then there is a duplicate IP address on the Ethernet network similar to that of the local module.

	If the LED is **green (flashing)**, then the module has booted and is running correctly **without** any application configuration loaded. If the LED is **green (solid)**, then the module has booted and is running correctly **with** application configuration loaded.
A / B	The Ethernet LED will light up when an Ethernet link has been detected (by plugging in a connected Ethernet cable). The LED will flash every time traffic is detected. This module has two Ethernet ports A and B. Each LEDs represents each specific port.
Act	The Activity LED will flash each time there is activity with any BACnet device on any interface (BACnet/IP, BACnet Ethernet, or MSTP). Flashing Red – A failed transaction has occurred on one of the BACnet devices (e.g., corrupted packet, unknown packet, no response, etc.). Flashing Green – A successful transaction has occurred on one of the BACnet devices. Off – No activity.
Ser	The Serial Port LED indicates the activity on the MSTP RS485 Port. Flashing Red – A corrupted or incorrect BACnet packet was received. Flashing Green – A valid BACnet packet was received. Off – No BACnet packets are being received or sent.
Aux	The Aux LED will flash each time there is activity on any of the primary interfaces (e.g., EtherNet/IP Target Class 1 communication). Flashing Red – A corrupted or incorrect packet was received on one of the Primary Interfaces (EtherNet/IP, Modbus TCP). Flashing Green – A valid packet was received on one of the Primary Interfaces (EtherNet/IP, Modbus TCP). Off – No activity.

Table 8.1 - Module LED operation

8.2. MODULE STATUS MONITORING IN SLATE

The BACnet Master provides various statistics which can assist with module operation, maintenance, and fault finding. The statistics can be accessed in full by Slate or using the web server in the module.

To view the module's status in the Aparian-Slate environment, the module must be online. If the module is not already Online (following a recent configuration download), then right-click on the module and select the ***Go Online*** option.

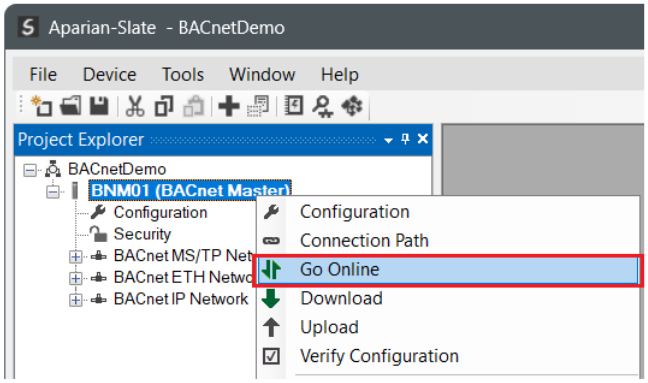


Figure 8.2 - Selecting to Go Online

The Online mode is indicated by the green circle behind the module in the Project Explorer tree.

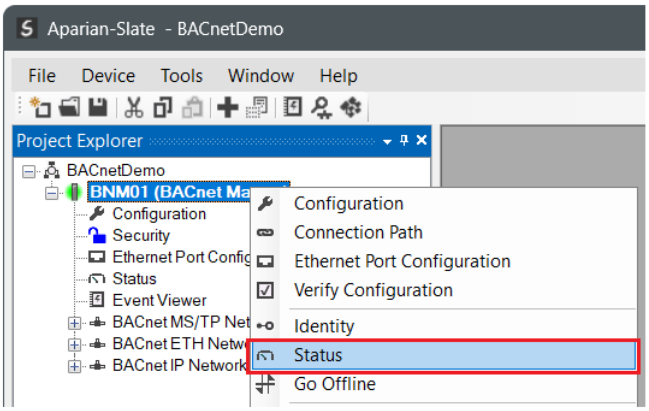


Figure 8.3 - Selecting online Status

The Status monitoring window can be opened by either double-clicking on the ***Status*** item in the Project Explorer tree, or by right-clicking on the module and selecting ***Status***. The status window contains multiple tabs to display the current status of the module.

8.2.1. GENERAL

The General tab displays the general status for the local BACnet Master module.

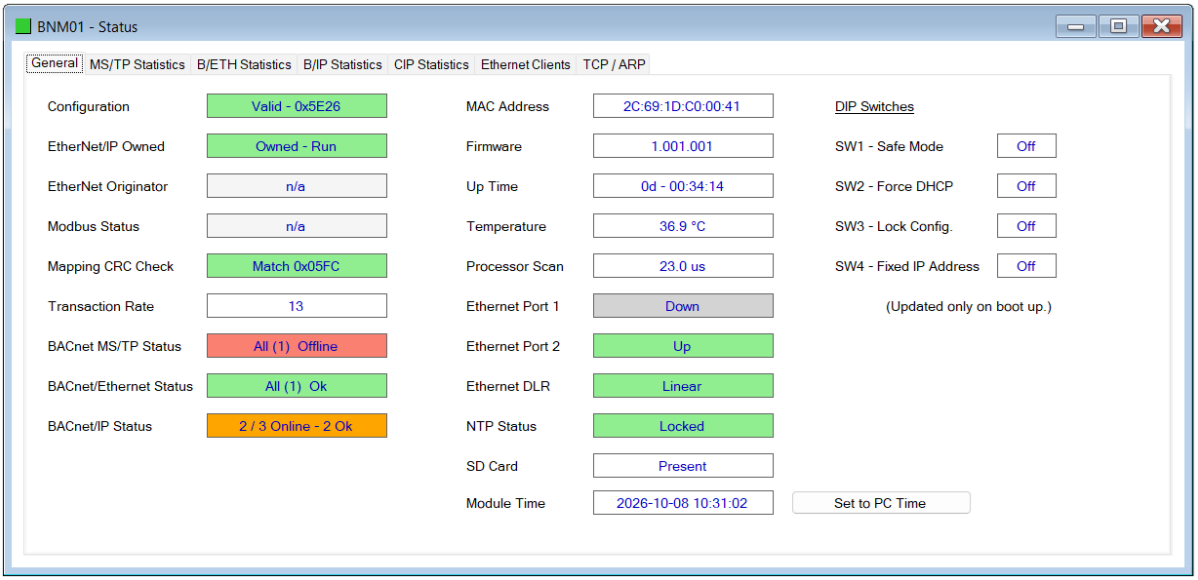


Figure 8.4 - Status monitoring – General

The General tab displays the following general parameters:

Parameter	Description
Configuration	Indicates whether the current configuration is valid or not, as well as the configuration checksum.
EtherNet/IP Owned	When the module is configured as an EtherNet/IP Target, this will indicate if the module is owned by an EtherNet/IP connection originator, as well as the Run/Program state of the controller.
EtherNet/IP Originator	When the module is configured as an EtherNet/IP Originator, this will show if all the Class 1 and Explicit Message connections to EtherNet/IP target devices are established and returning valid data.
Modbus Status	When the module is operating as a **Modbus Server**, this parameter will indicate that the module has received a valid Modbus request within the Modbus inactivity time. When the module is operating as a **Modbus Client**, this parameter will indicate that all the mapping items in the Modbus Auxiliary Map are executing correctly.
Transaction Rate	Number of BACnet transactions executed per second.
BACnet MS/TP Status	The status of the BACnet MS/TP Network. Shows the number of configured devices that are Online and in Data Exchange.
BACnet /Ethernet Status	The status of the BACnet /Ethernet Network. Shows the number of configured devices that are Online and in Data Exchange.
BACnet /IP Status	The status of the BACnet IP Network.

	Shows the number of configured devices that are Online and in Data Exchange.
MAC Address	Displays the module's unique Ethernet MAC address.
Firmware	The version of the module's firmware.
Up Time	Indicates the elapsed time since the module was powered-up.
Temperature	The internal temperature of the module's CPU.
Processor Scan	The amount of time (microseconds) taken by the module's processor in the last scan.
Ethernet Port 1/2	This is the status of each Ethernet port. **Down** The Ethernet connector has **not been** successfully connected to an Ethernet network. **Up** The Ethernet connector has successfully connected to an Ethernet network. **Mirror Enabled** The Ethernet port is mirroring the traffic on the other Ethernet port.
Ethernet DLR (Device Level Ring)	The status of the Ethernet DLR. **Disabled** Device Level Ring functionality has been disabled. **Linear** The DLR functionality has been enabled and the Ethernet network architecture is linear. **Ring – Fault** The DLR functionality has been enabled and the Ethernet network architecture is ring, but there is a fault with the network. **Ring – Ok** The DLR functionality has been enabled and the Ethernet network architecture is ring and is operating as expected.
NTP Status	The status of the local NTP Client. **Disabled** The NTP time synchronization has been disabled. **Locked** NTP time synchronization has been enabled and the module has locked onto the target time server. **Not Locked** NTP time synchronization has been enabled and the module has not locked onto the target time server.
SD Card	Indicates if a SD Card is present or not.
Module Time	The current module time.
DIP Switch Position	The status of the DIP switches when the module booted.

	NOTE: This status will not change if the DIP switches are altered when the module is running.

Table 8.2 - Parameters displayed in the Status Monitoring – General Tab

8.2.2. BACnet MS/TP Statistics

The BACnet MS/TP Statistics tab displays the statistics associated with the BACnet MS/TP communication network.

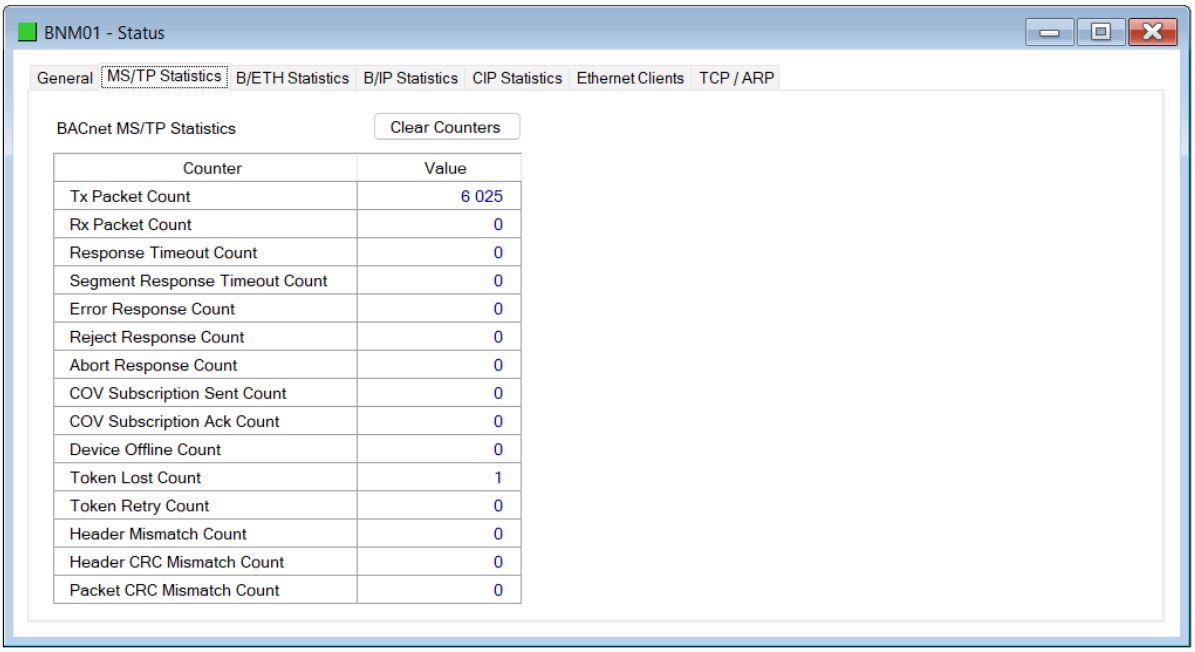


Figure 8.5 - Status monitoring – BACnet MS/TP Statistics

Statistic	Description
Tx Packet Count	The number of packets transmitted.
Rx Packet Count	The number of packets received.
Response Timeout Count	The number of requests that did not receive a valid response within the configured time-out period.
Segment Response Timeout Count	The number of segmented responses where the subsequent segment was not received within the required time-out period.
Error Response Count	The number of requests where the device responded with an Error response.
Reject Response Count	The number of requests where the device responded with a Reject response.
Abort Response Count	The number of requests where the device responded with an Abort response.

COV Subscription Sent Count	The number of COV (Change-Of-Value) subscription registration requests sent.
COV Subscription Ack Count	The number of COV (Change-Of-Value) subscription registration acknowledgments received.
Device Offline Count	The number of occurrences the device went offline.
Token Lost Count	The number of times the token was passed, but lost by the receiving node.
Token Retry Count	The number of times the token had to be resent, because the other receiving device did not use or receive the received token.
Header Mismatch Count	The number of received packets where the header format was incorrect.
Header CRC Mismatch Count	The number of received packets where the header checksum failed.
Packet CRC Mismatch Count	The number of received packets where the packet payload checksum failed.

Table 8.3 – BACnet MS/TP statistics

8.2.3. BACnet /Ethernet Statistics

The BACnet /Ethernet Statistics tab displays the statistics associated with the BACnet /Ethernet communication network.

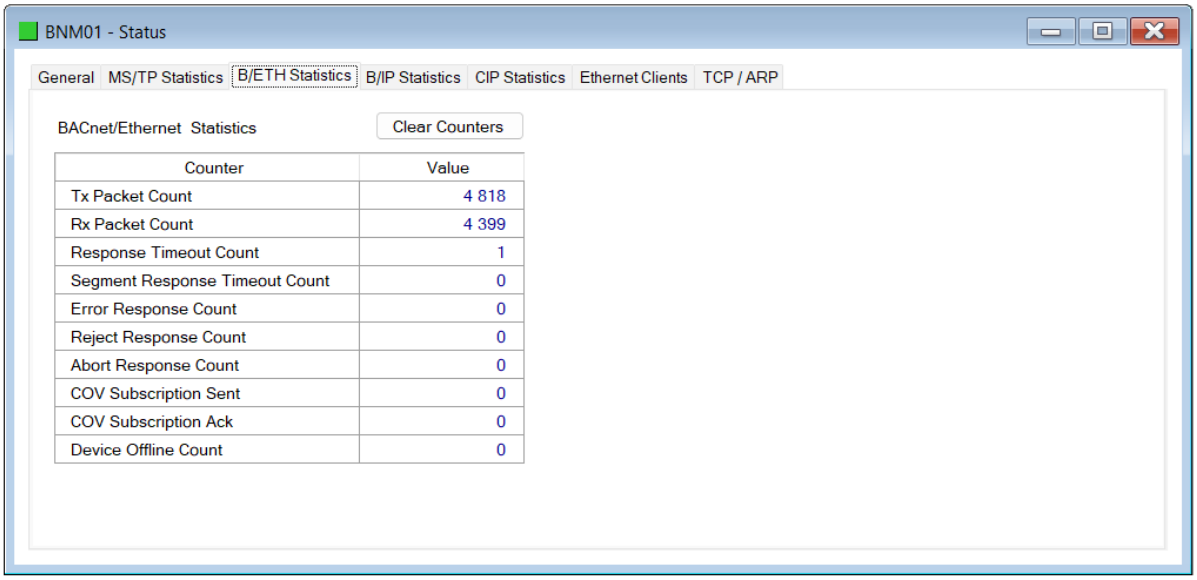


Figure 8.6 - Status monitoring – BACnet /Ethernet Statistics

Statistic	Description
Tx Packet Count	The number of packets transmitted.
Rx Packet Count	The number of packets received.
Response Timeout Count	The number of requests that did not receive a valid response within the configured time-out period.
Segment Response Timeout Count	The number of segmented responses where the subsequent segment was not received within the required time-out period.
Error Response Count	The number of requests where the device responded with an Error response.
Reject Response Count	The number of requests where the device responded with a Reject response.
Abort Response Count	The number of requests where the device responded with an Abort response.
COV Subscription Sent Count	The number of COV (Change-Of-Value) subscription registration requests sent.
COV Subscription Ack Count	The number of COV (Change-Of-Value) subscription registration acknowledgments received.
Device Offline Count	The number of occurrences the device went offline.

Table 8.4 – BACnet /Ethernet statistics

8.2.4. BACnet /IP Statistics

The BACnet /IP Statistics tab displays the statistics associated with the BACnet /IP communication network.

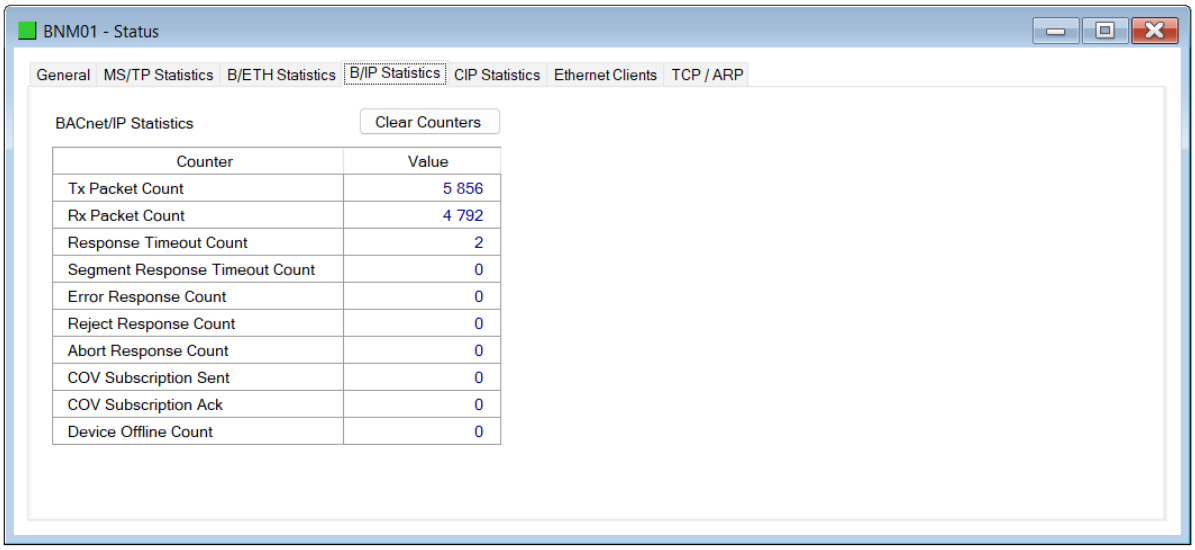


Figure 8.7 - Status monitoring – BACnet /IP Statistics

Statistic	Description
Tx Packet Count	The number of packets transmitted.
Rx Packet Count	The number of packets received.
Response Timeout Count	The number of requests that did not receive a valid response within the configured time-out period.
Segment Response Timeout Count	The number of segmented responses where the subsequent segment was not received within the required time-out period.
Error Response Count	The number of requests where the device responded with an Error response.
Reject Response Count	The number of requests where the device responded with a Reject response.
Abort Response Count	The number of requests where the device responded with an Abort response.
COV Subscription Sent Count	The number of COV (Change-Of-Value) subscription registration requests sent.
COV Subscription Ack Count	The number of COV (Change-Of-Value) subscription registration acknowledgments received.
Device Offline Count	The number of occurrences the device went offline.

Table 8.5 – BACnet /IP statistics

8.2.5. EtherNet/IP Explicit

The EtherNet/IP Explicit Statistics tab displays the statistics associated with EtherNet/IP Device explicit mapping.

NOTE: This tab is only applicable when the module has the Primary Interface set to EtherNet/IP Originator.

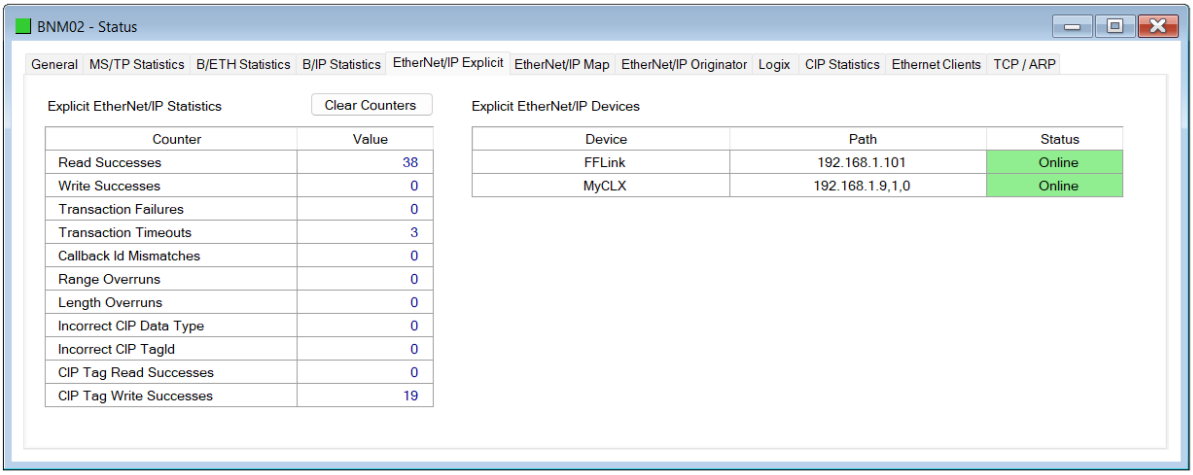


Figure 8.8 - Status monitoring – EtherNet/IP Explicit

Statistic	Description
Read Successes	The number of successful reads from the target EtherNet/IP device.
Write Successes	The number of successful writes to the target EtherNet/IP device.
Transaction Failures	The number of failed reads/writes to the target EtherNet/IP device (e.g. error response).
Transaction Timeouts	The number of times the target EtherNet/IP device failed to respond.
Callback Id Mismatches	The EtherNet/IP UCMM or Class 3 response does not match the request.
Range Overruns	The number of times the returned data amount runs over the maximum Internal Data Space.
Length Overruns	The number of times the returned data is greater than the configured get length.
Incorrect CIP Data Type	When the Explicit Message Function is a Tag Read/Write, this statistic will increase when the incorrect CIP data type was returned in the response.
Incorrect CIP Tag Id	When the Explicit Message Function is a Tag Read/Write, this statistic will increase when the incorrect CIP UDT tag ID was returned in the response.
CIP Tag Read Successes	When the Explicit Message Function is a Tag Read, this statistic will increase when there was a successful Logix Tag Read.
CIP Tag Write Successes	When the Explicit Message Function is a Tag Write, this statistic will increase when there was a successful Logix Tag Write.

Table 8.6 – EtherNet/IP Explicit Statistics

8.2.6. EtherNet/IP Map

The EtherNet/IP Map tab displays the success counts for each EtherNet/IP device mapped item.

NOTE: This tab is only applicable when the module has the Primary Interface set to EtherNet/IP Originator.

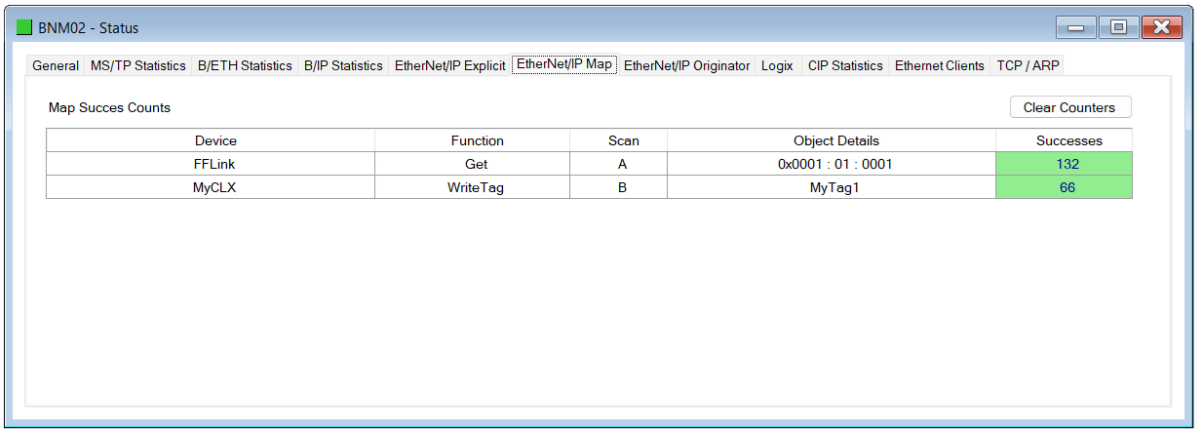


Figure 8.9 - Status monitoring – EtherNet/IP Map

Each time a mapped item is executed successfully its associated count will increase. The count cell will momentarily be highlighted green following a successful transaction.

8.2.7. EtherNet/IP Originator

The EtherNet/IP Originator tab displays the EtherNet/IP Class 1 connection status and statistics for each configured EtherNet/IP device.

NOTE: This tab is only applicable when the module has the Primary Interface set to EtherNet/IP Originator.

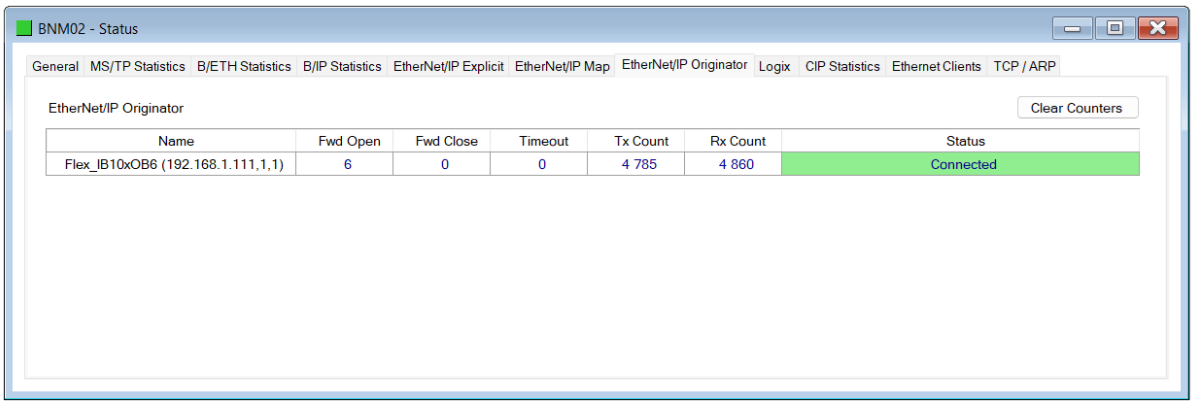


Figure 8.10 - Status monitoring – EtherNet/IP Originator

Statistic	Description
Status	The current connection status of the module. **Connected** The device is connected and exchanging data using Class 1 cyclic communication. **Offline** The device is offline and not connected **Various response faults** If the connection parameters entered are not correct, then generally the target device will reply with the specific reason for the connection rejection, for example: Ownership Conflict Connection In Use Or Duplicate Forward Open
Class 1 Originator Statistics	
Forward Open Count	The number of Class 1 Forward Open (connection establishment) messages sent to this device.
Forward Close Count	The number of Class 1 Forward Close (connection termination) messages sent or received from this device.
Connection Timeouts	The number of this connection was closed due to timeouts.
Tx Count	Number of Class 1 messages sent to the specific target device.
Rx Count	Number of Class 1 messages received from the specific target device.

Table 8.7 – EtherNet/IP Class 1 status and statistics

8.2.8. LOGIX

The Logix tab displays the Logix statistics for the explicit EtherNet/IP Tag Read/Write message instructions.

NOTE: This tab is only relevant when the module has the Primary Interface set to EtherNet/IP Originator and Logix Tag Read/Write functions are being used in the EtherNet/IP Explicit Message Map.

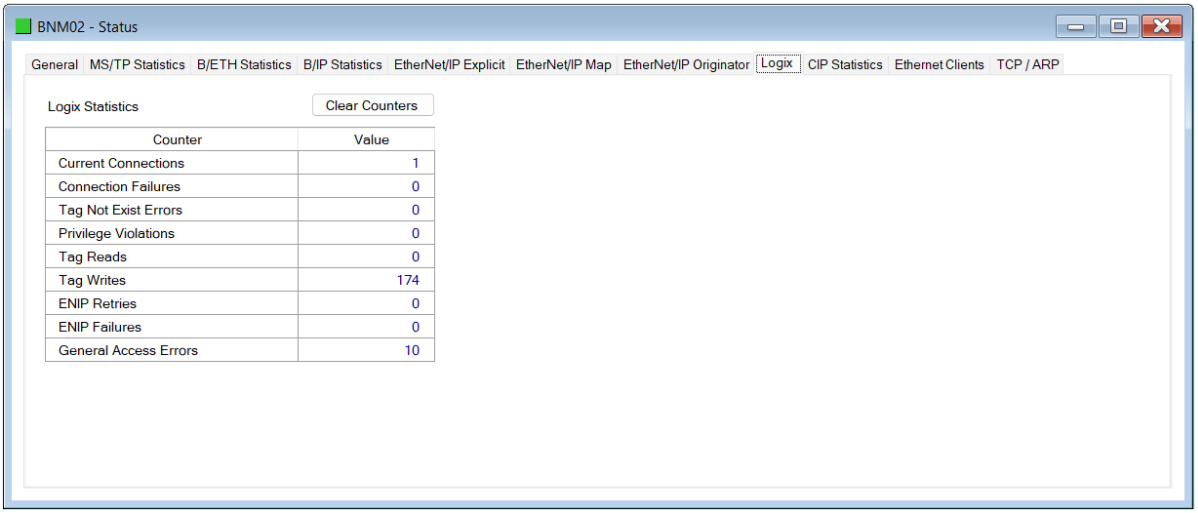


Figure 8.11 - Status monitoring – Logix Statistics

Parameter	Description
Current Connections	The number of current open class 3 connections.
Connection Failures	The number of failed attempts at establishing a class 3 connection with a Logix controller.
Tag Not Exist Errors	The number of tag read and tag write transactions that failed due to the destination tag not existing.
Privilege Violations	The number of tag read and tag write transactions that failed due to a privilege violation error. Note: This may be caused by the ***External Access*** property of the Logix tag being set to either ***None*** or ***Read Only***.
Tag Reads	The number of tag read transactions executed by the BACnet Master module.
Tag Writes	The number of tag write transactions executed by the BACnet Master module.
ENIP Retries	This count increases when no response is received from the Logix Controller within the ENIP timeout.
ENIP Failures	This count increases when the ENIP Retry Limit is reached and no response has been received from the Logix Controller.
Access General Error	This count increases when a tag cannot be accessed for any other reason not reported above.

Table 8.8 – Logix Statistics Tab

8.2.9. MODBUS

The Modbus tab displays the Modbus statistics for the Modbus Read and Write Message Exchanges when the module is a Modbus Server or Modbus Client.

NOTE: The Modbus statistics tab is only displayed if the module has the primary interface set to Modbus Client or Modbus Server.

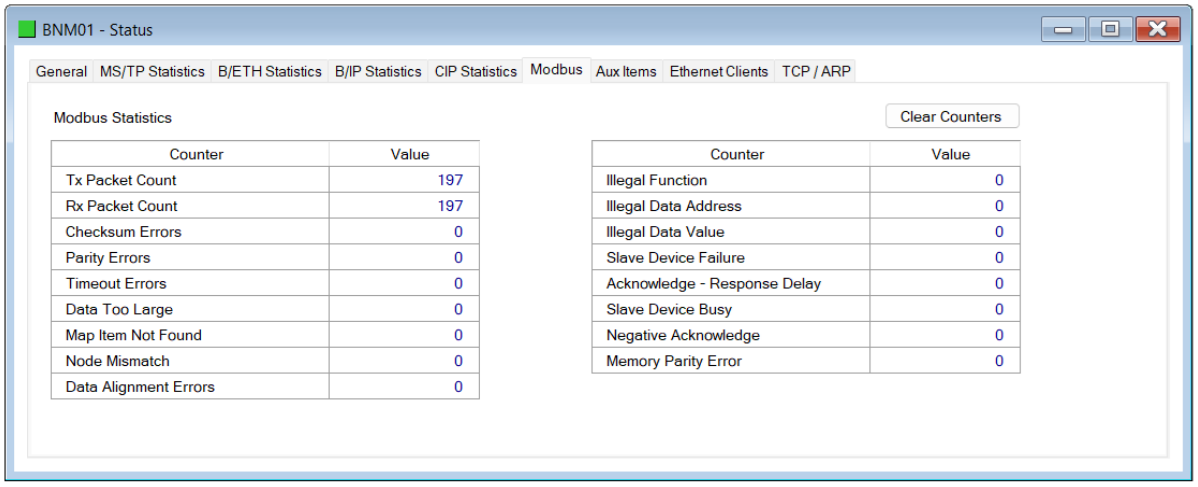


Figure 8.12. - Status monitoring – Modbus Statistics

The Modbus tab displays the following parameters:

Statistic	Description
Tx Packet Count	The number of Modbus packets sent by the module.
Rx Packet Count	The number of Modbus packets received by the module.
Checksum errors	The number of corrupted Modbus packets received by the module.
Parity errors	The number of bytes with parity errors received by the module.
Timeout Errors	The number of message response timeouts the module has encountered.
Data Too Large	The number of Modbus requests or responses where the data was too large to process.
Map Item Not Found	The number of Modbus requests did not match any mapped items.
Node Mismatch	The received Modbus request did not match the module's Modbus node address.
Data Alignment Errors	The Modbus request and associated mapped item is not byte aligned with the destination Logix tag.
Illegal Function	The number of times the Modbus device responded with an Illegal Function exception.

Illegal Data Address	The number of times the Modbus device responded with an Illegal Data Address exception.
Illegal Data Value	The number of times the Modbus device responded with an Illegal Data Value exception.
Slave Device Failure	The number of times the Modbus device responded with a Device Failure exception.
Acknowledge –Response Delay	The number of times the Modbus device responded with an Acknowledge exception.
Slave Device Busy	The number of times the Modbus device responded with a Slave Busy exception.
Negative Acknowledge	The number of times the Modbus device responded with a Negative Acknowledge exception.
Memory Parity Error	The number of times the Modbus device responded with a Memory Parity exception.

Table 8.9 - Modbus Statistics Tab

8.2.10. MODBUS AUXILIARY ITEMS

The Modbus Auxiliary Items tab will display the status of the last transaction by each mapping item configured in the Modbus Auxiliary Map. If the last transaction failed, then the status will show **Fail** and if it was successful then it will show **Ok**. If the transaction shows **Waiting**, then it indicates that this mapped item has a trigger and the mapped item has not yet been executed. This provides quick visual feedback as to which items are currently active.

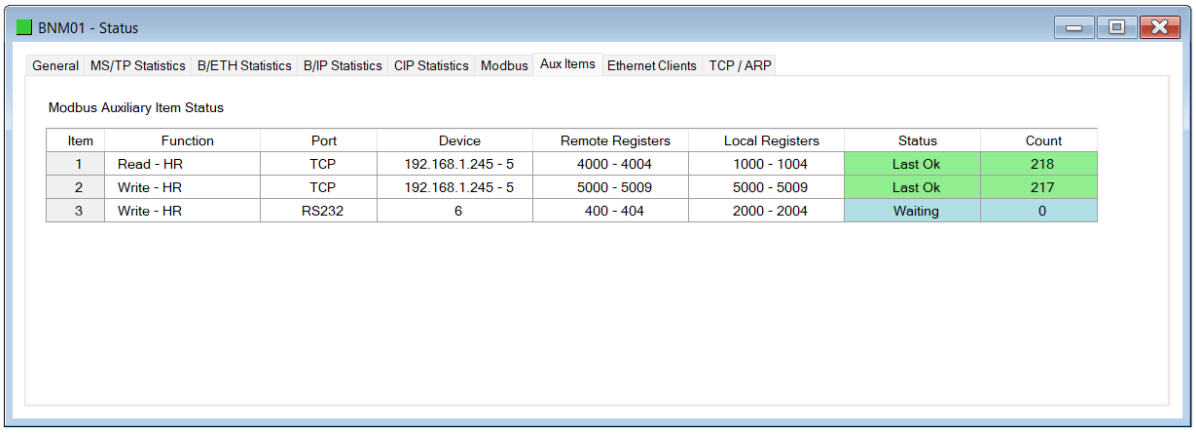


Figure 8.13 - Auxiliary Item status.

8.2.11. CIP STATISTICS

The CIP tab displays the Ethernet CIP statistics.

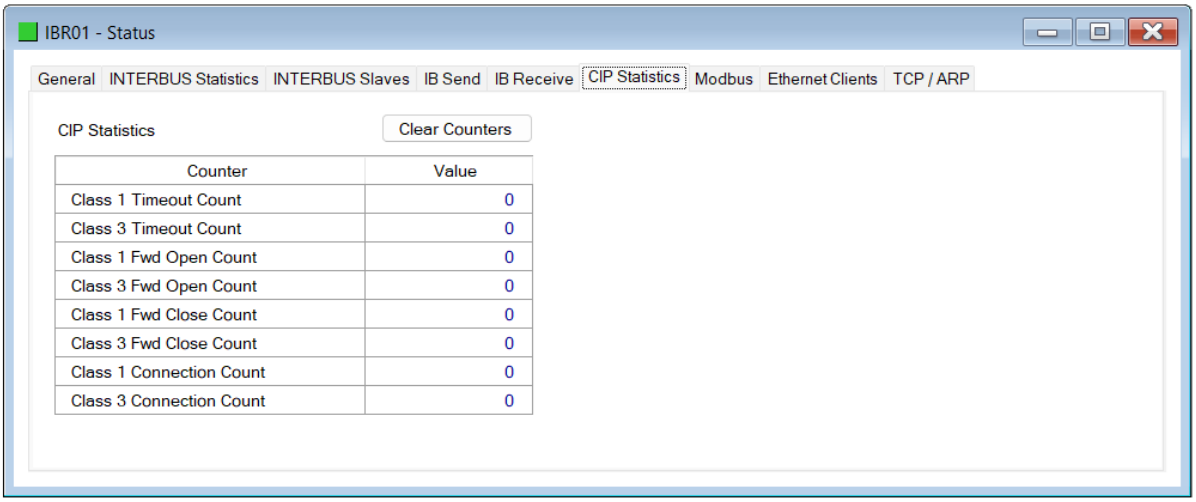


Figure 8.14 - Status monitoring – CIP Statistics

Statistic	Description
Class 1 Timeout Count	The number of Class 1 connections closed due to Timeouts.
Class 3 Timeout Count	The number of Class 3 connections closed due to Timeouts.
Class 1 Forward Open Count	The number of Class 1 Forward Open (connection establishment) messages sent.
Class 3 Forward Open Count	The number of Class 3 Forward Open (connection establishment) messages sent.
Class 1 Forward Close Count	The number of Class 1 Forward Close (connection termination) messages sent.
Class 3 Forward Close Count	The number of Class 3 Forward Close (connection termination) messages sent.
Class 1 Connection Count	The current number of active Class 1 connections.
Class 3 Connection Count	The current number of active Class 3 connections.

Table 8.10 – Mapped Item statistics

8.2.12. ETHERNET CLIENTS

The Ethernet Clients tab displays details of the Ethernet and EtherNet/IP clients connected to the BACnet Master.

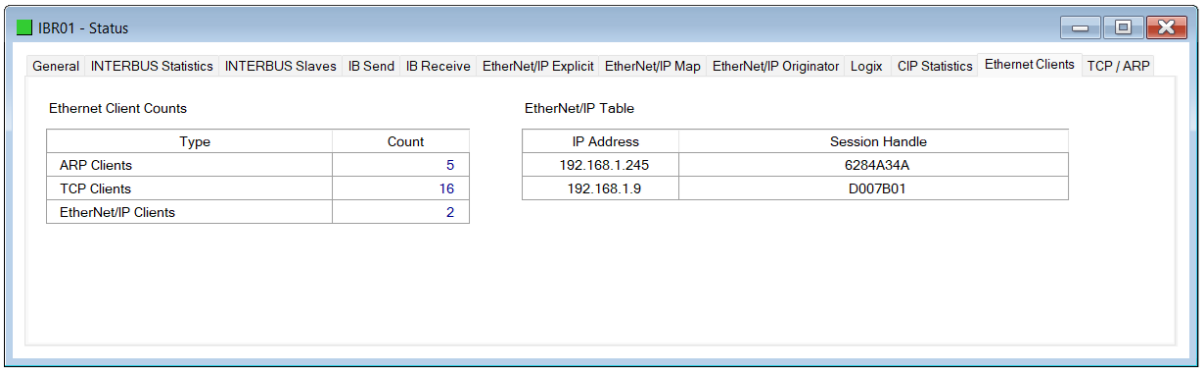


Figure 8.15 –Status monitoring – Ethernet Client Statistics

8.2.13. TCP/ARP

The TCP/ARP tab displays details of the internal Ethernet ARP and TCP lists of the BACnet Master.

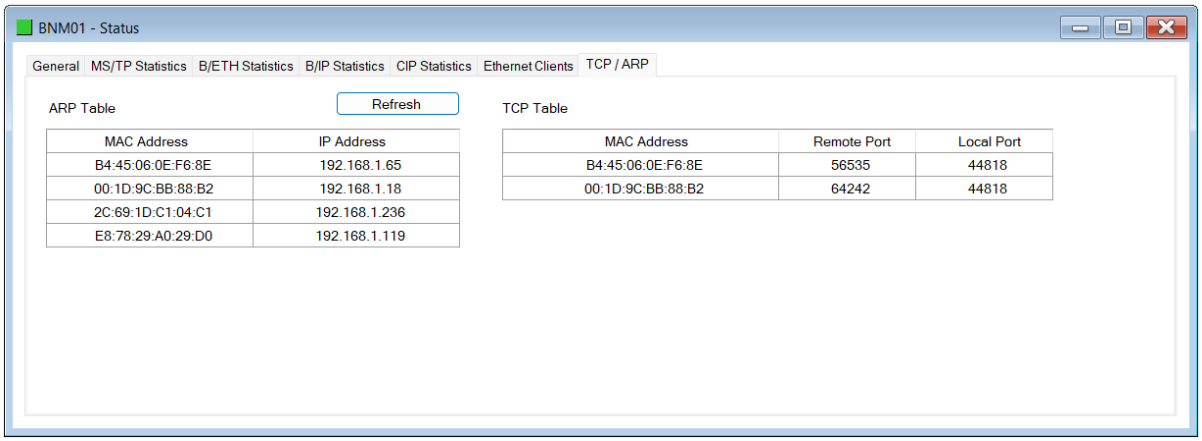


Figure 8.16 – Status monitoring – Ethernet TCP / ARP Statistics

NOTE: The Refresh button must be clicked to refresh the ARP table.

8.3. ETHERNET/IP TARGET DEVICE STATUS MONITORING IN SLATE

When online with the module in Slate, right-click on the desired EtherNet/IP device under the *EtherNet/IP Connections* tree in Slate and select *Status*.

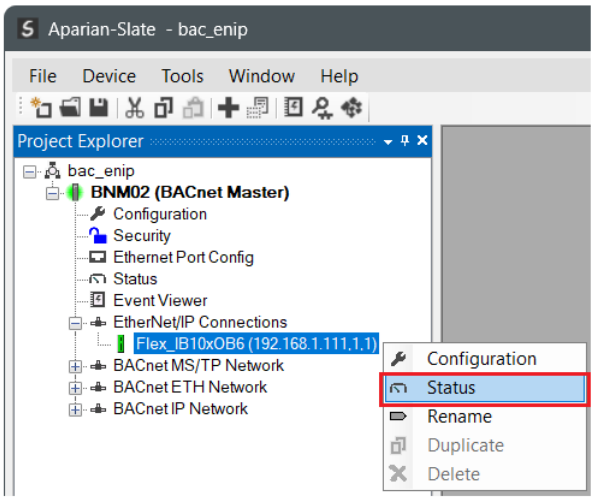


Figure 8.17 – EtherNet/IP Device Status – Status selection

8.3.1.1. GENERAL

The General Status for the EtherNet/IP device shows the connection statistics and parameters associated with the EtherNet/IP Class 1 connection.

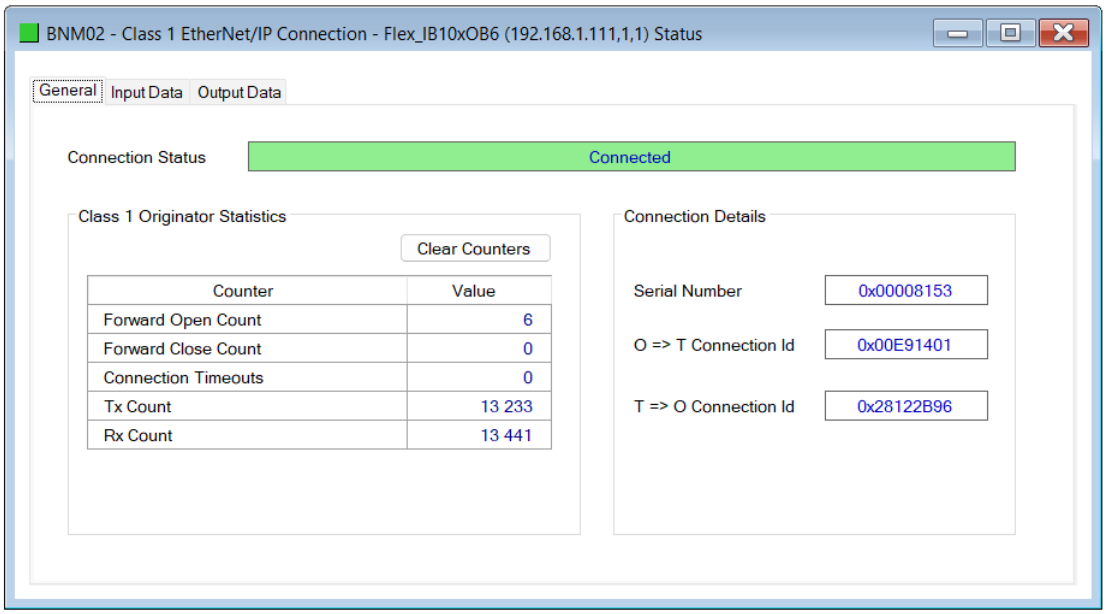


Figure 8.18 – EtherNet/IP Device Status – General Status

Statistic	Description
Connection Status	The current connection status of the module. **Connected** The device is connected and exchanging data using Class 1 cyclic communication. **Offline**

	The device is offline and not connected **Various response faults** If the connection parameters entered are not correct, then generally the target device will reply with the specific reason for the connection rejection, for example: Connection Status: Invalid Originator To Target Size
Class 1 Originator Statistics	
Forward Open Count	The number of Class 1 Forward Open (connection establishment) messages sent to this device.
Forward Close Count	The number of Class 1 Forward Close (connection termination) messages sent or received from this device.
Connection Timeouts	The number of this connection was closed due to timeouts.
Tx Count	Number of Class 1 messages sent to the specific target device.
Rx Count	Number of Class 1 messages received from the specific target device.
Connection Details	
Serial Number	The active connection's serial number.
O -> T Connection Id	The active connection Originator to Target Connection Id.
T -> O Connection Id	The active connection Target to Originator Connection Id.

Table 8.11 – EtherNet/IP Class 1 Device status and statistics

8.3.1.2. INPUT DATA

The Input Data for the EtherNet/IP device shows the Input Assembly associated with the EtherNet/IP Class 1 connection.

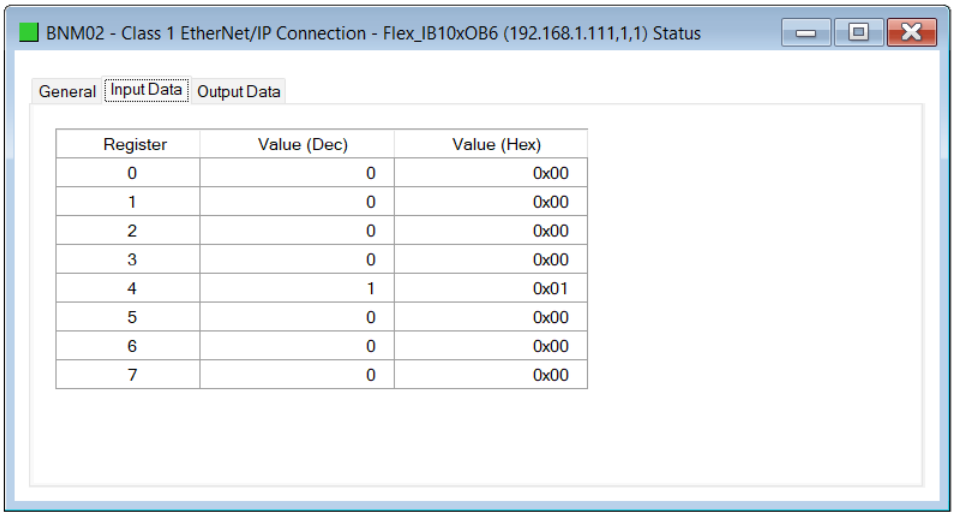


Figure 8.19 – EtherNet/IP Device Status – Input Data

8.3.1.3. OUTPUT DATA

The Output Data for the EtherNet/IP device shows the Output Assembly associated with the EtherNet/IP Class 1 connection.

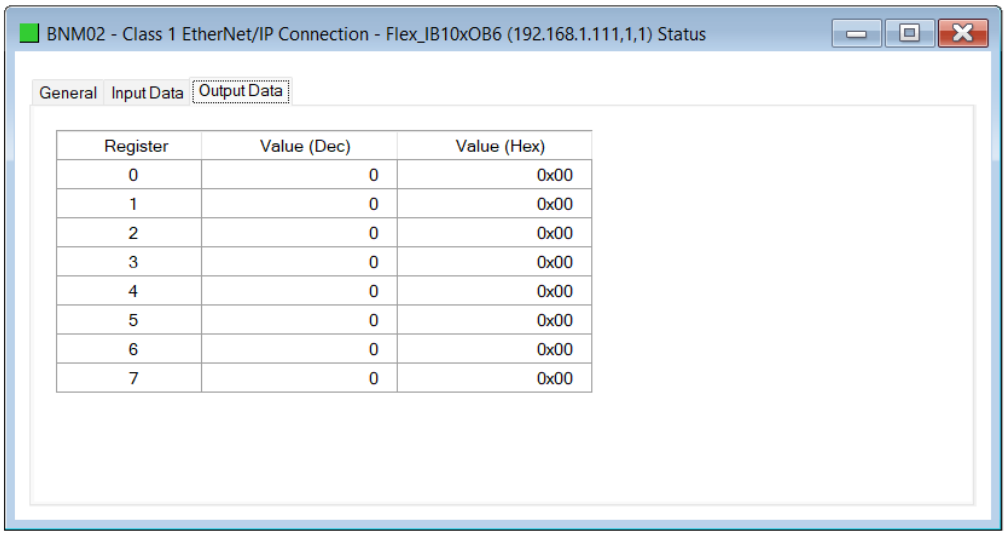


Figure 8.20 – EtherNet/IP Device Status – Output Data

8.4. BACNET DEVICE MONITORING

When online with the module in Slate, right-click on the desired BACnet device under one of the BACnet Networks and select *Status*.

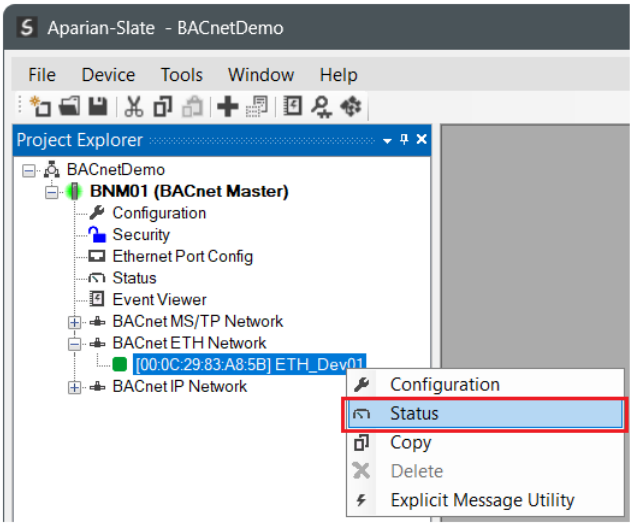


Figure 8.21 – BACnet Device Status – Status selection

8.4.1.1. GENERAL

The General Status for the BACnet device shows the device status and a number of communication statistics.

The Status will reflect whether the device is online, and whether it successfully in Data Exchange. The latter is implied by all properties read/writes executing successfully.

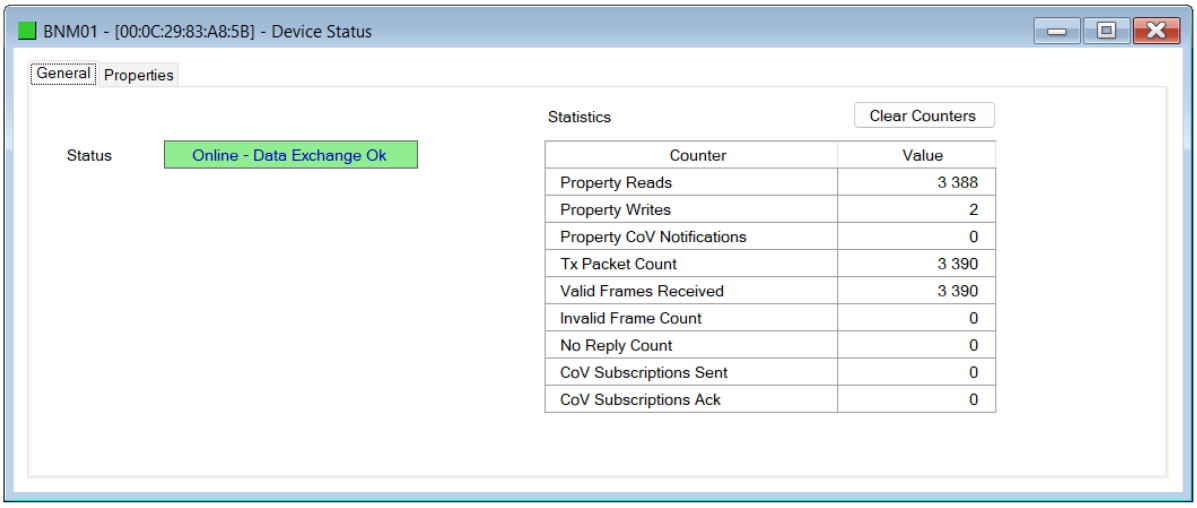


Figure 8.22 – BACnet Device Status – General Status

The following BACnet device statistics are displayed:

Statistic	Description
Property Reads	The number of successful property read transactions.
Property Writes	The number of successful property write transactions.
Property CoV Notifications	The number of property CoV notifications received.
Tx Packet Count	The number of packets transmitted.
Valid Frames Received	The number of valid packets received.
Invalid Frame Count	The number of invalid or malformed packets received.
No Reply Count	The number of packets sent that received no reply within the required time-out period.
COV Subscription Sent Count	The number of COV (Change-Of-Value) subscription registration requests sent.
COV Subscription Ack Count	The number of COV (Change-Of-Value) subscription registration acknowledgments received.

Table 8.12 – BACnet device statistics

8.4.1.1. PROPERTIES

The Properties Status for the BACnet device shows the (expanded) configured device properties and their latest status.

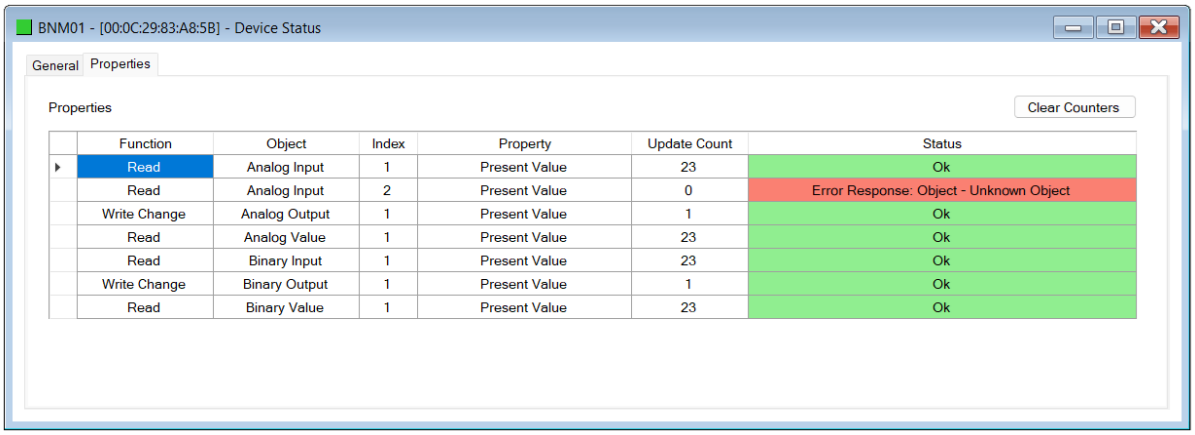


Figure 8.23 – BACnet Device Status – Properties Status

The following BACnet device statistics are displayed:

Statistic	Description
Property Reads	The number of successful property read transactions.
Property Writes	The number of successful property write transactions.
Property CoV Notifications	The number of property CoV notifications received.
Tx Packet Count	The number of packets transmitted.
Valid Frames Received	The number of valid packets received.
Invalid Frame Count	The number of invalid or malformed packets received.
No Reply Count	The number of packets sent that received no reply within the required time-out period.
COV Subscription Sent Count	The number of COV (Change-Of-Value) subscription registration requests sent.
COV Subscription Ack Count	The number of COV (Change-Of-Value) subscription registration acknowledgments received.

Table 8.13 – BACnet device statistics

8.5. MODULE EVENT LOG

The BACnet Master module logs various diagnostic records to an internal event log. These logs are stored in non-volatile memory and can be displayed using Slate or via the web interface. To view them in Slate, select the ***Event Viewer*** option in the Project Explorer tree.

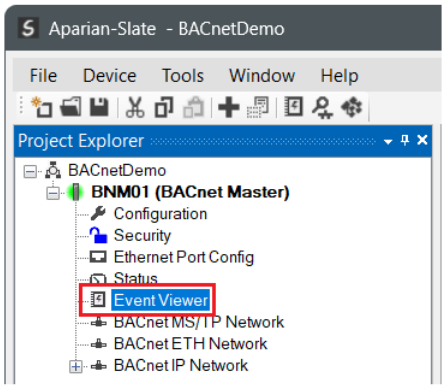
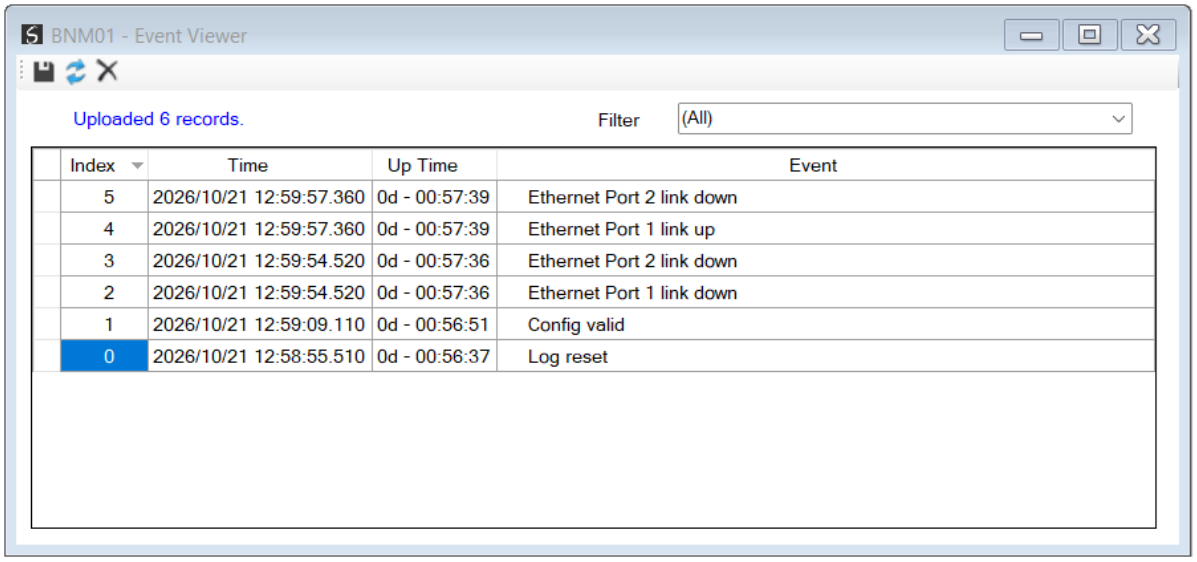


Figure 8.24 - Selecting the module Event Log

The Event Log window will open and automatically read all the events from the module. The log entries are sorted so as to have the latest record at the top. Custom sorting is achieved by double-clicking on the column headings.



Index	Time	Up Time	Event
5	2026/10/21 12:59:57.360	0d - 00:57:39	Ethernet Port 2 link down
4	2026/10/21 12:59:57.360	0d - 00:57:39	Ethernet Port 1 link up
3	2026/10/21 12:59:54.520	0d - 00:57:36	Ethernet Port 2 link down
2	2026/10/21 12:59:54.520	0d - 00:57:36	Ethernet Port 1 link down
1	2026/10/21 12:59:09.110	0d - 00:56:51	Config valid
0	2026/10/21 12:58:55.510	0d - 00:56:37	Log reset

Figure 8.25 - Module Event Log

The log can also be stored to a file for future analysis, by selecting the Save button in the tool menu. To view previously saved files, use the Event Log Viewer option under the Tools menu.

8.6. BACNET PACKET CAPTURE

The BACnet Master module provides the capability to capture BACnet traffic for analysis and to assist with remote support.

To start a BACnet capture right-click on the required BACnet network and select the ***BACnet Packet Capture*** menu option.

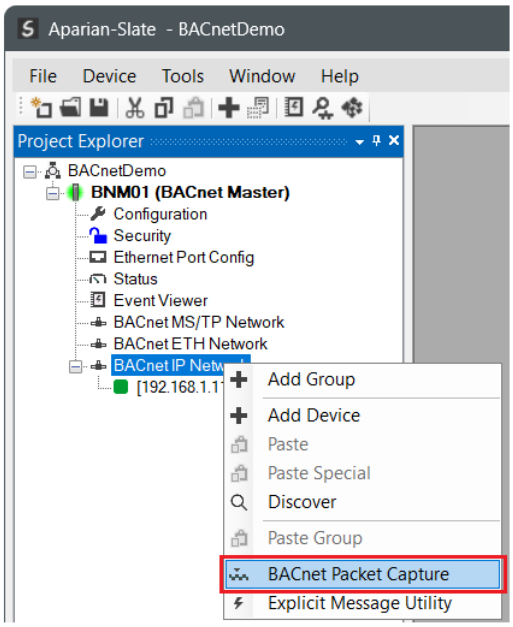


Figure 8.26 - Selecting BACnet Packet Capture

The BACnet Packet Capture window will open and automatically start capturing all the relevant packets.

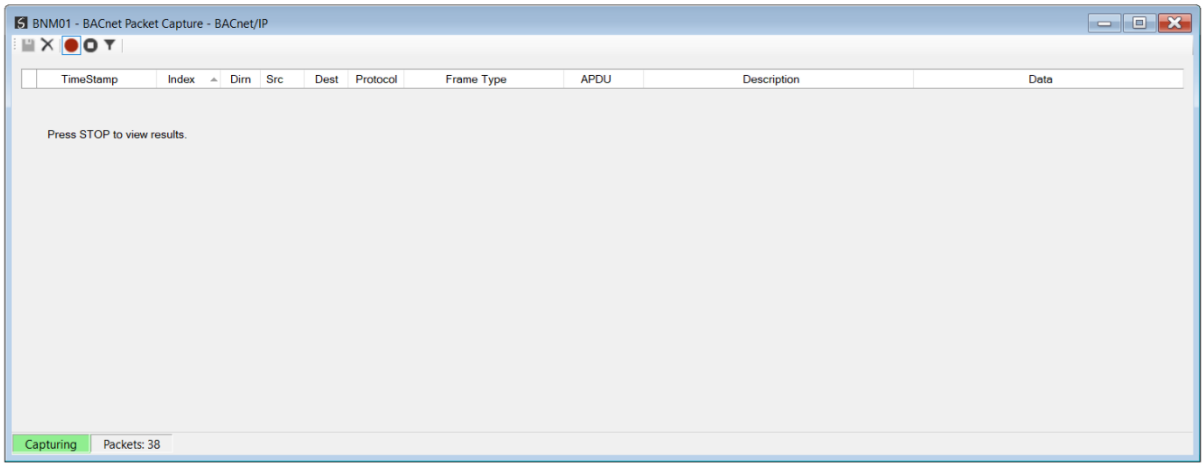
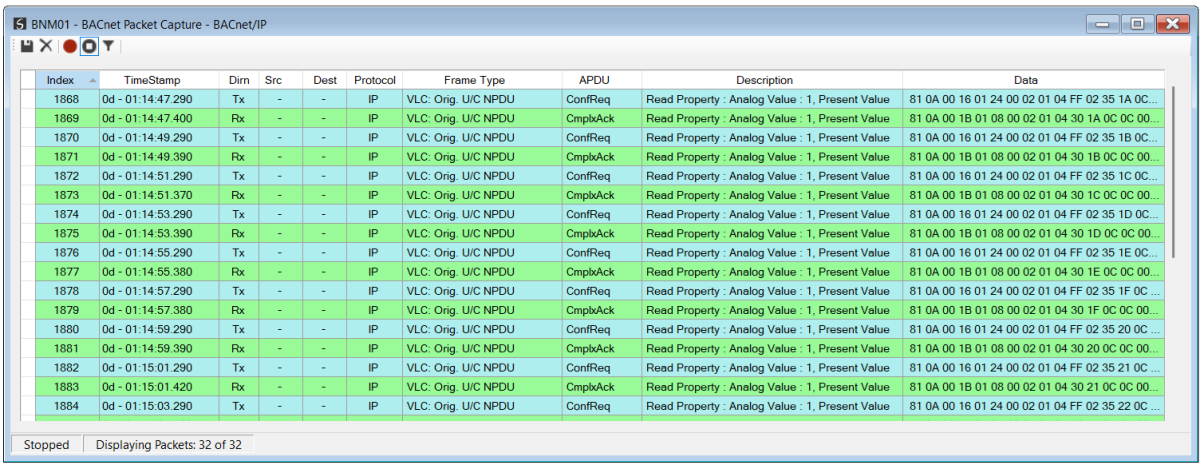


Figure 8.27 – BACnet packet capture started

To display the captured packets, the capture process must first be stopped, by pressing the ***Stop*** button.



Index	TimeStamp	Dirn	Src	Dest	Protocol	Frame Type	APDU	Description	Data
1868	0d - 01:14:47.290	Tx	-	-	IP	VLC: Orig. U/C NPDU	ConfReq	Read Property : Analog Value : 1, Present Value	81 0A 00 16 01 24 00 02 01 04 FF 02 35 1A 0C...
1869	0d - 01:14:47.400	Rx	-	-	IP	VLC: Orig. U/C NPDU	CmplxAck	Read Property : Analog Value : 1, Present Value	81 0A 00 1B 01 08 00 02 01 04 30 1A 0C 0C 00...
1870	0d - 01:14:49.290	Tx	-	-	IP	VLC: Orig. U/C NPDU	ConfReq	Read Property : Analog Value : 1, Present Value	81 0A 00 16 01 24 00 02 01 04 FF 02 35 1B 0C...
1871	0d - 01:14:49.390	Rx	-	-	IP	VLC: Orig. U/C NPDU	CmplxAck	Read Property : Analog Value : 1, Present Value	81 0A 00 1B 01 08 00 02 01 04 30 1B 0C 0C 00...
1872	0d - 01:14:51.290	Tx	-	-	IP	VLC: Orig. U/C NPDU	ConfReq	Read Property : Analog Value : 1, Present Value	81 0A 00 16 01 24 00 02 01 04 FF 02 35 1C 0C...
1873	0d - 01:14:51.370	Rx	-	-	IP	VLC: Orig. U/C NPDU	CmplxAck	Read Property : Analog Value : 1, Present Value	81 0A 00 1B 01 08 00 02 01 04 30 1C 0C 0C 00...
1874	0d - 01:14:53.290	Tx	-	-	IP	VLC: Orig. U/C NPDU	ConfReq	Read Property : Analog Value : 1, Present Value	81 0A 00 16 01 24 00 02 01 04 FF 02 35 1D 0C...
1875	0d - 01:14:53.390	Rx	-	-	IP	VLC: Orig. U/C NPDU	CmplxAck	Read Property : Analog Value : 1, Present Value	81 0A 00 1B 01 08 00 02 01 04 30 1D 0C 0C 00...
1876	0d - 01:14:55.290	Tx	-	-	IP	VLC: Orig. U/C NPDU	ConfReq	Read Property : Analog Value : 1, Present Value	81 0A 00 16 01 24 00 02 01 04 FF 02 35 1E 0C...
1877	0d - 01:14:55.380	Rx	-	-	IP	VLC: Orig. U/C NPDU	CmplxAck	Read Property : Analog Value : 1, Present Value	81 0A 00 1B 01 08 00 02 01 04 30 1E 0C 0C 00...
1878	0d - 01:14:57.290	Tx	-	-	IP	VLC: Orig. U/C NPDU	ConfReq	Read Property : Analog Value : 1, Present Value	81 0A 00 16 01 24 00 02 01 04 FF 02 35 1F 0C ...
1879	0d - 01:14:57.380	Rx	-	-	IP	VLC: Orig. U/C NPDU	CmplxAck	Read Property : Analog Value : 1, Present Value	81 0A 00 1B 01 08 00 02 01 04 30 1F 0C 0C 00...
1880	0d - 01:14:59.290	Tx	-	-	IP	VLC: Orig. U/C NPDU	ConfReq	Read Property : Analog Value : 1, Present Value	81 0A 00 16 01 24 00 02 01 04 FF 02 35 20 0C ...
1881	0d - 01:14:59.390	Rx	-	-	IP	VLC: Orig. U/C NPDU	CmplxAck	Read Property : Analog Value : 1, Present Value	81 0A 00 1B 01 08 00 02 01 04 30 20 0C 0C 00...
1882	0d - 01:15:01.290	Tx	-	-	IP	VLC: Orig. U/C NPDU	ConfReq	Read Property : Analog Value : 1, Present Value	81 0A 00 16 01 24 00 02 01 04 FF 02 35 21 0C ...
1883	0d - 01:15:01.420	Rx	-	-	IP	VLC: Orig. U/C NPDU	CmplxAck	Read Property : Analog Value : 1, Present Value	81 0A 00 1B 01 08 00 02 01 04 30 21 0C 0C 00...
1884	0d - 01:15:03.290	Tx	-	-	IP	VLC: Orig. U/C NPDU	ConfReq	Read Property : Analog Value : 1, Present Value	81 0A 00 16 01 24 00 02 01 04 FF 02 35 22 0C ...

Figure 8.28 – BACnet Packet Capture complete

The captured BACnet packets are tabulated as follows:

Statistic	Description
Index	The packet index, incremented for each packet sent or received.
Time	The elapsed time since the module powered up.
Dirn	The direction of the packet, either transmitted (Tx) or received (Rx).
Src	The Source Node address of the packet. (MS/TP only)
Dest	The Destination Node address of the packet. (MS/TP only)
Protocol	The BACnet protocol
Frame Type	The lower level BACnet frame type
APDU	The application layer protocol data unit type
Description	Description of the packet that was received.
Data	The raw packet data.

Table 8.14 – BACnet Packet Capture fields

The packet capture can be saved to a file for further analysis, by selecting the ***Save*** button on the toolbar. Previously saved BACnet Packet Capture files can be viewed by selecting the ***BACnet Packet Capture Viewer*** option in the ***Tools*** menu.

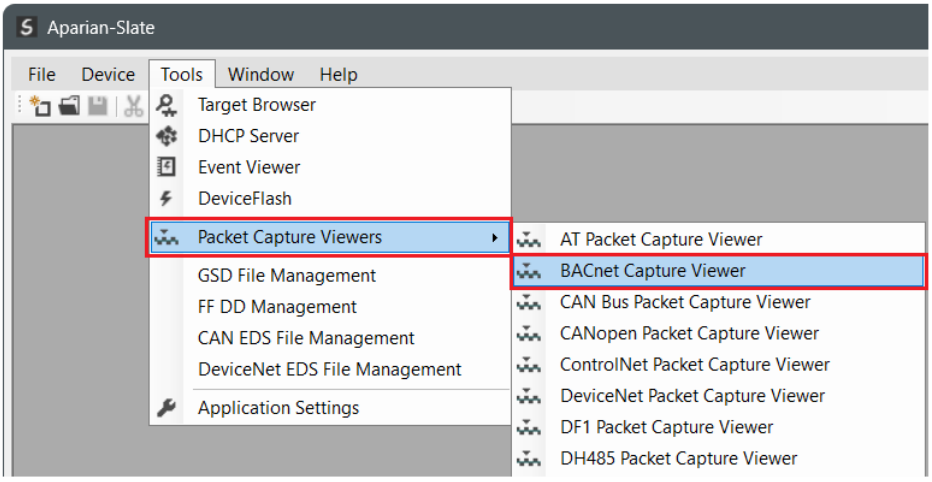


Figure 8.29 - Selecting the BACnet Packet Capture Viewer

8.7. BACNET EXPLICIT MESSAGE UTILITY

Slate provides a utility to manually read and write properties in a BACnet device. This utility can be accessed from a device in the configuration, or from the device Discover form.

To open the utility, right-click on the selected BACnet device and select the ***Explicit Message Utility*** option.

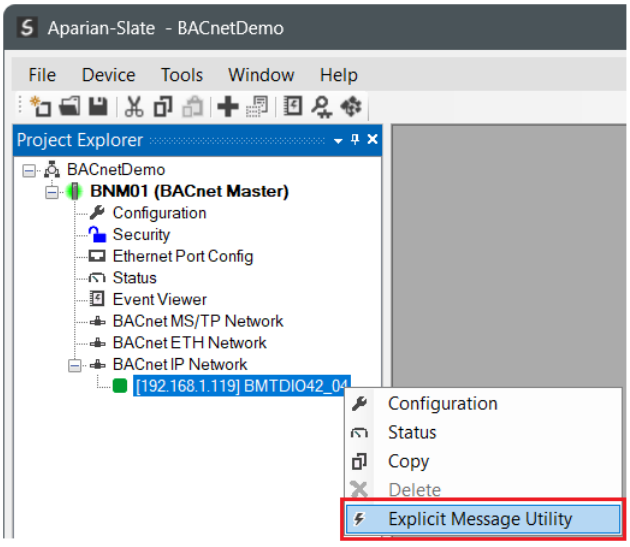


Figure 8.30 - Selecting the BACnet Explicit Message Utility

The utility will open and have all the network parameters pre-configured for that device. The user can then select the required ***Object Type***, ***Object Instance***, ***Property***, and ***Array Index*** if applicable. To initiate a read command, click on the ***Read*** button. The result will be displayed in the ***Result*** textbox.

To write to a property, first select the ***Write (Data) Type***, and enter the data in the ***Write Data*** textbox, then click the ***Write*** button.

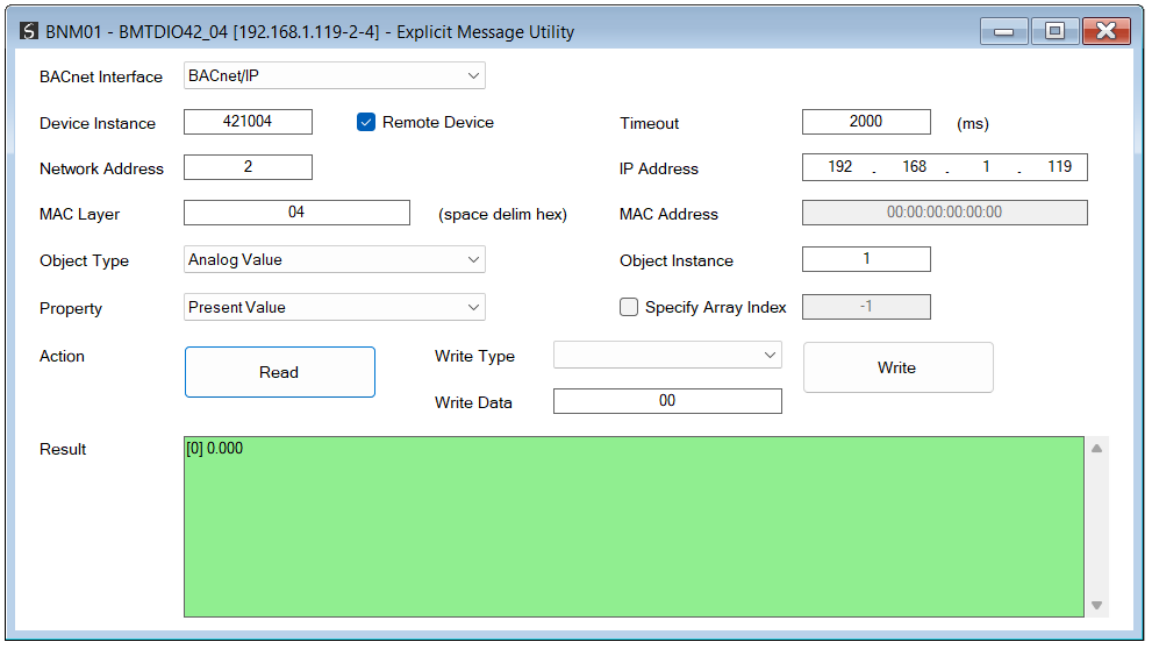


Figure 8.31 - BACnet Explicit Message Utility

8.8. MODBUS PACKET CAPTURE

The module provides the capability to capture Modbus traffic for analysis and to assist with remote support. To start a Modbus capture right-click on the module and select the ***Modbus Packet Capture*** menu option.

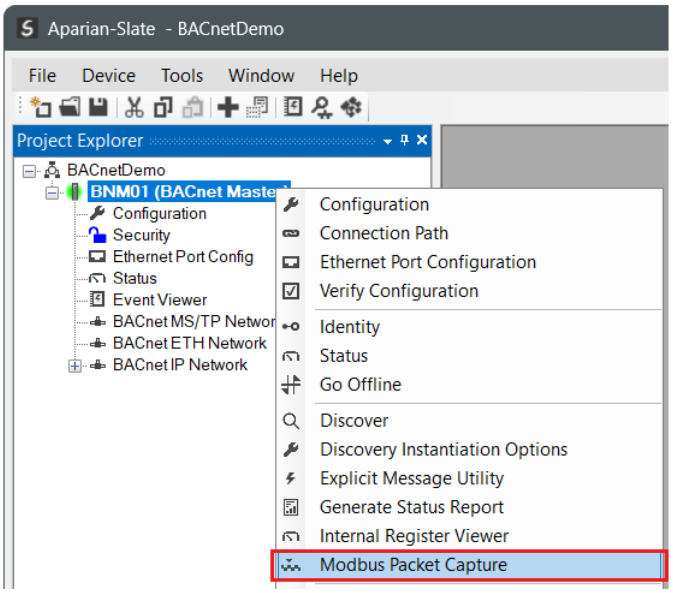


Figure 8.32 - Selecting Modbus Packet Capture

The Modbus Packet Capture window will open and automatically start capturing all Modbus packets.

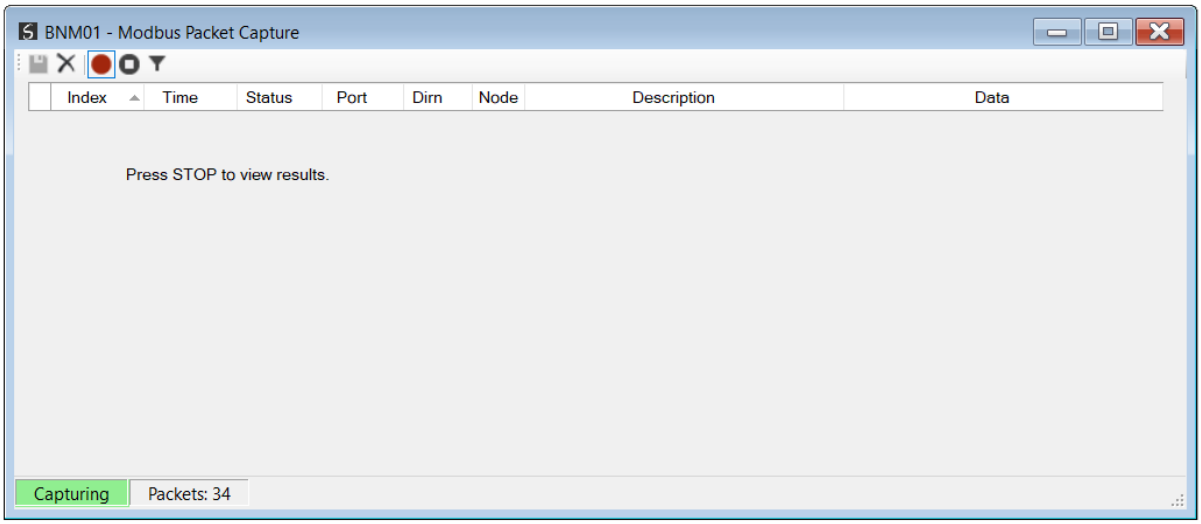


Figure 8.33 – Modbus packet capture

To display the captured Modbus packets, the capture process must first be stopped, by pressing the Stop button.

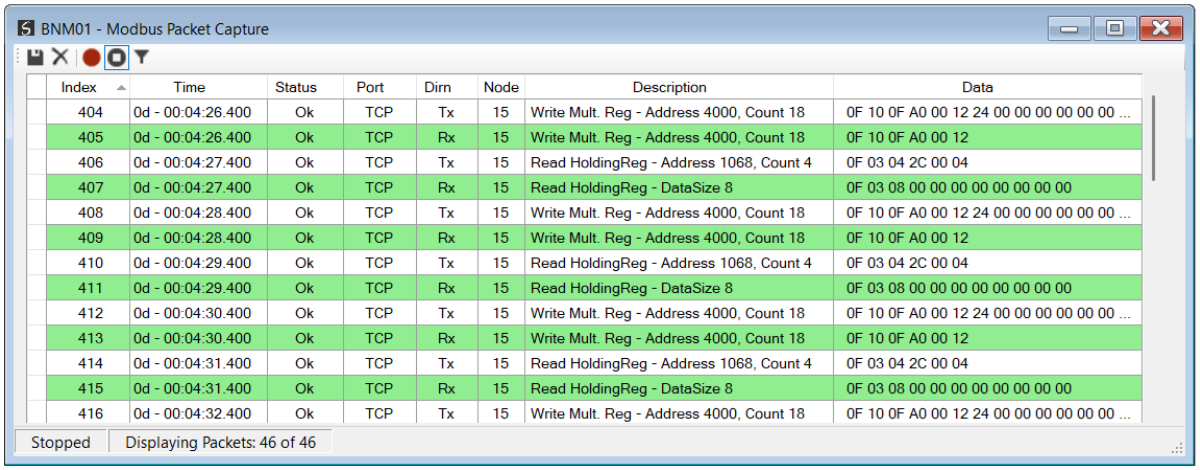


Figure 8.34 – Modbus Packet Capture complete

The captured Modbus packets are tabulated as follows:

Statistic	Description
Index	The packet index, incremented for each packet sent or received.
Time	The elapsed time since the module powered up.

Status	The status of the packet. Received packets are checked for valid Modbus constructs and valid checksums.
Port	Port on where the data was sent or received (TCP)
Dirn	The direction of the packet, either transmitted (Tx) or received (Rx).
Node	The Source Node address for the packet
Description	Description of the packet that was received.
Data	The raw packet data.

Table 8.15 – Modbus Packet Capture fields

The packet capture can be saved to a file for further analysis, by selecting the ***Save*** button on the toolbar. Previously saved Modbus Packet Capture files can be viewed by selecting the ***Modbus Packet Capture Viewer*** option in the ***Tools*** menu.

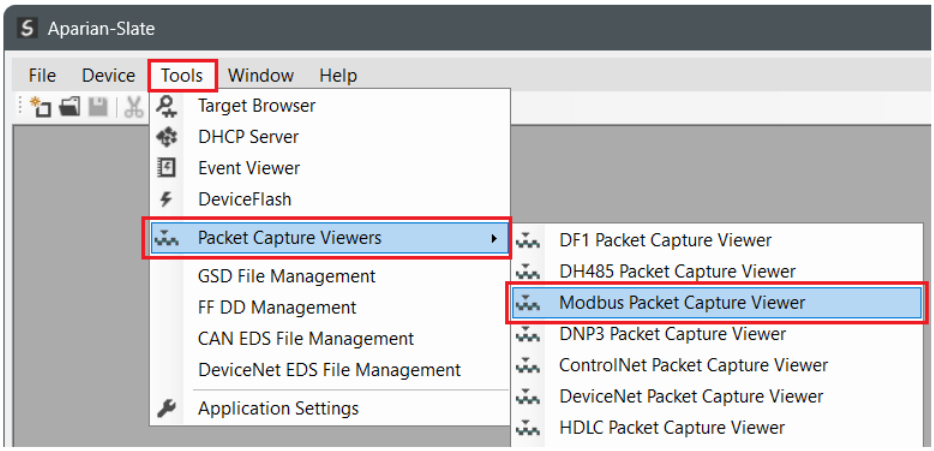


Figure 8.35 - Selecting the Modbus Packet Capture Viewer

8.9. MODULE STATUS REPORT

For assisting with support Slate can generate a status report for the module which is a Microsoft Word compatible document that can be emailed to support. To generate this report the user can right-click on the module (when online in Slate) and select *Generate Status Report*.

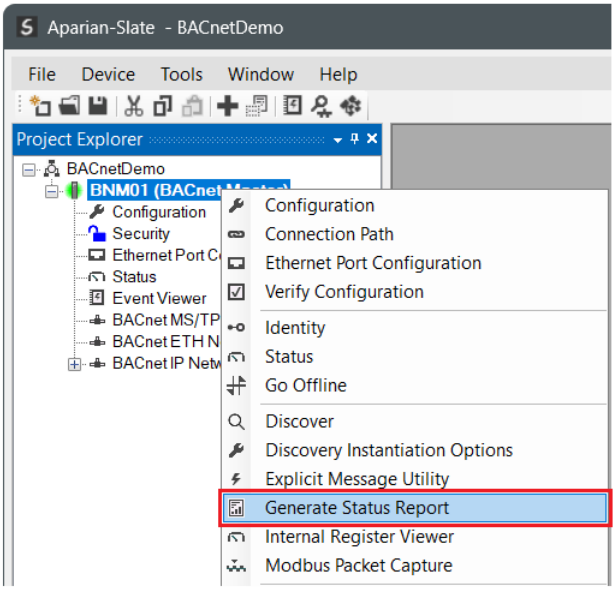


Figure 8.36 – Module Status Report

8.10. MODBUS SUMMARY CSV

Slate can provide a summary of the Modbus registers being used in the form of a CSV file. This will assist the user to better understand where what data is being mapped (based on the BACnet Master module configuration). To generate the Modbus Summary CSV, right-click on the module (when online in Slate), and select *Export Modbus Summary CSV*.

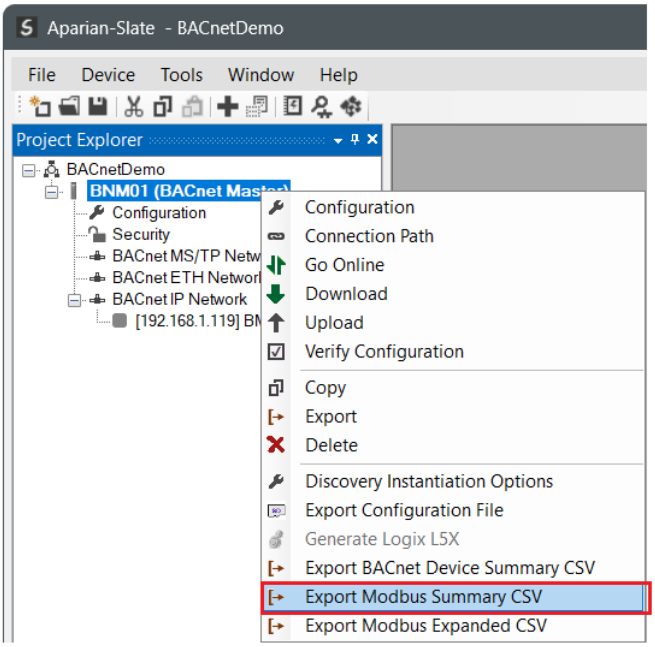


Figure 8.37 – Modbus Summary Report Generation

	A	B	C	D	E	F
1	Reg Type	Modbus Function	Start Address	End Address	Element Count	Description
2	HR	Read	1000	1047	48	System.Status
3	HR	Write	1048	1049	2	System.Control
4	HR	Read	1050	1051	2	BACnet Device.BMTDIO42_04.Status
5	HR	Read	1052	1055	4	BACnet Device.BMTDIO42_04.Data
6	(End)					

Figure 8.38 – Modbus Summary CSV

8.11. MODBUS EXPANDED CSV

Slate can provide a detailed version of the Modbus registers being used in the form of a CSV file. This will assist the user to better understand where what data is being mapped (based on the BACnet Master module configuration). To generate the Modbus Expanded CSV, right-click on the module (when online in Slate), and select *Export Modbus Expanded CSV*.

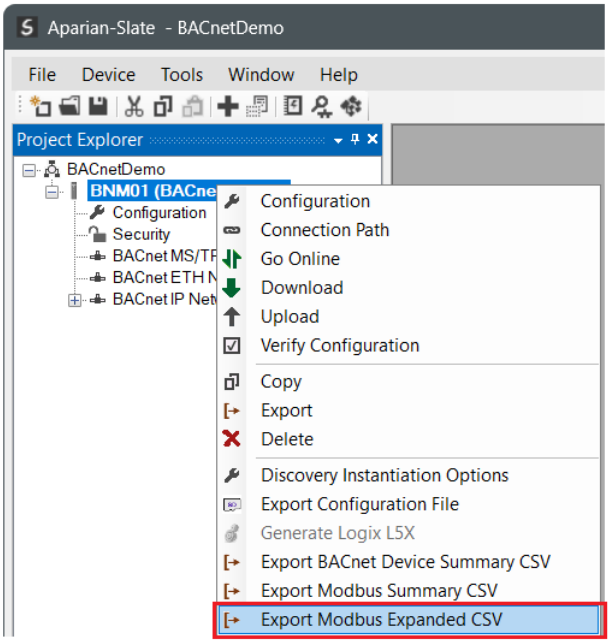


Figure 8.39 – Modbus Expanded Report Generation

	A	B	C	D	E	F
1	Reg Type	Modbus Function	Start Address	End Address	Element Count	Description
2	HR	Read	1000	1000	1	System.Status.GeneralStatus
3	HR	Read	1002	1002	1	System.Status.ConfigCRC
4	HR	Read	1003	1003	1	System.Status.TransactionRate
5	HR	Read	1004	1004	1	System.Status.BACNet MSTPDevicesOnlineCount
6	HR	Read	1005	1005	1	System.Status.BACNet ETHDevicesOnlineCount
7	HR	Read	1006	1006	1	System.Status.BACNet IPPDevicesOnlineCount
8	HR	Read	1007	1007	1	System.Status.Reserved1
9	HR	Read	1008	1008	1	System.Status.BACNet MSTP DevicesExchangeCount
10	HR	Read	1009	1009	1	System.Status.BACNet ETH DevicesExchangeCount
11	HR	Read	1010	1010	1	System.Status.BACNet IP DevicesExchangeCount
12	HR	Read	1011	1011	1	System.Status.Reserved2
13	HR	Read	1012	1027	16	System.Status.DeviceOnline
14	HR	Read	1028	1043	16	System.Status.DeviceDataExchange
15	HR	Read	1044	1044	1	System.Status.ActualMappingCRC
16	HR	Read	1045	1045	1	System.Status.ReceivedMappingCRC
17	HR	Read	1046	1047	2	System.Status.Reserved
18	HR	Write	1048	1049	2	System.Control
19	HR	Read	1050	1050	1	BMTDIO42_04.Status.GeneralStatus
20	HR	Read	1051	1051	1	BMTDIO42_04.Status.PropertyUpdates
21	HR	Read	1052	1052	1	BMTDIO42_04.In.AV001PV.Status
22	HR	Read	1053	1053	1	BMTDIO42_04.In.AV001PV.SuccessCount
23	HR	Read	1054	1055	2	BMTDIO42_04.In.AV001PV
24	(End)					

Figure 8.40 – Modbus Expanded CSV

8.12. INTERNAL REGISTER VIEWER

The module provides an Internal Register Viewer that allows the user to view the data registers in the Internal Data Space (IDS) and Modbus Registers.

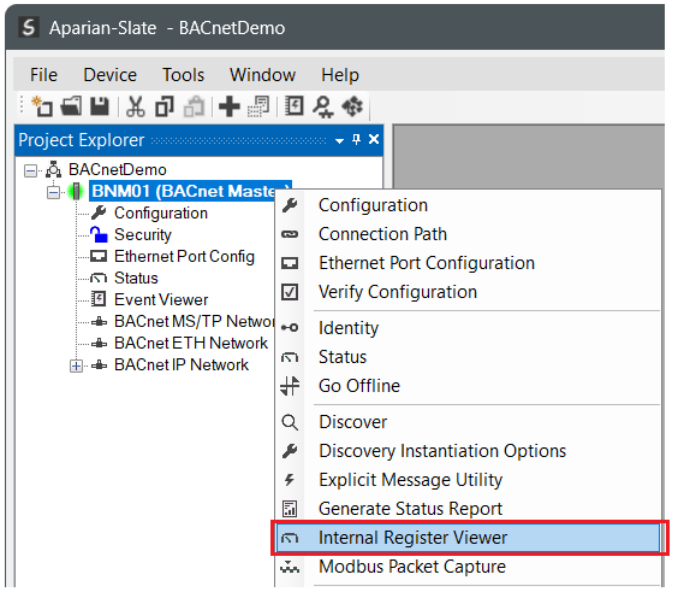


Figure 8.41 – Internal Register Viewer - Select

Once opened, the register type can be selected. Note that the Modbus Registers will only be available if the Primary Interface configured is for Modbus Server or Modbus Client.

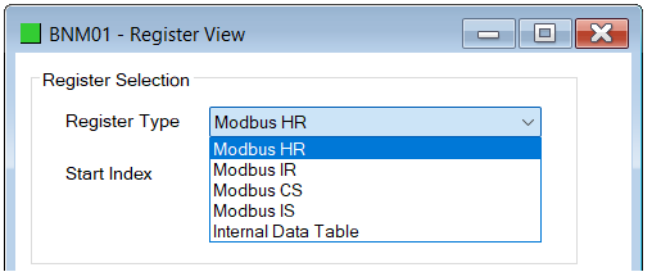


Figure 8.42 – Internal Register Viewer – Register Type

The Start and Stop Index can then be selected to determine how much data to display and refresh.

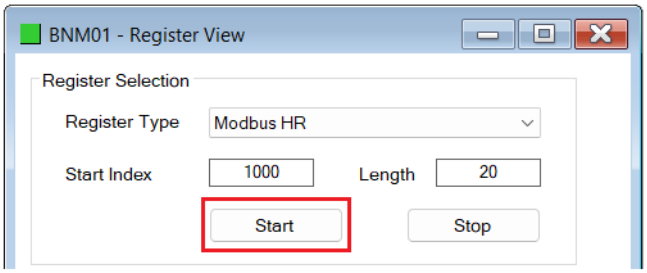


Figure 8.43 – Internal Register Viewer – Indexes

The data will appear and start refreshing by pressing the *Start* button.

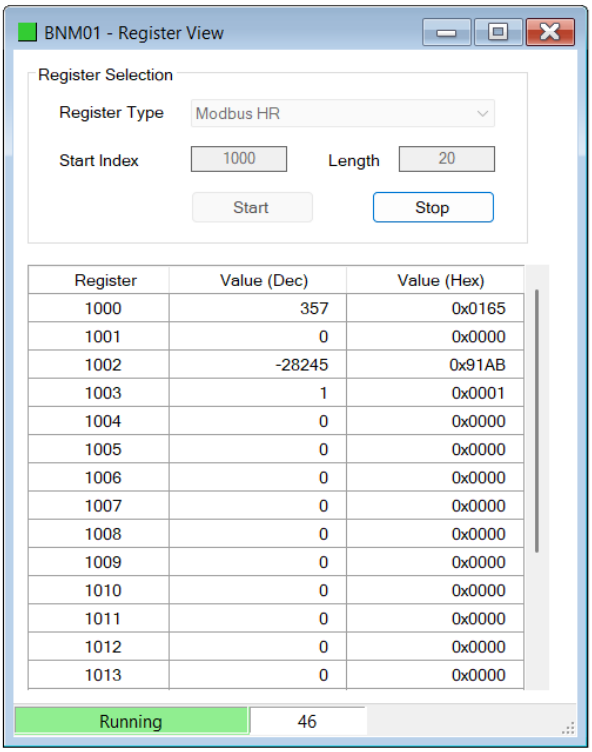


Figure 8.44 – Internal Register Viewer – Running

9. BACNET OBJECT PROPERTIES

9.1. SUPPORTED BACNET OBJECTS

The following BACnet objects are supported.

Object	*Object Name*
AI	Analog Input
AO	Analog Output
AV	Analog Value
BI	Binary Input
BO	Binary Output
BV	Binary Value
DV	Device
GR	Group
LSP	Life Safety Point
MI	Multi-State Input
MO	Multi-State Output
MV	Multi-State Value

Table 9.1 – Supported BACnet Objects

9.2. SUPPORTED BACNET OBJECT PROPERTIES

The following BACnet properties are supported for each BACnet object:

Object	*Property Name*	*BACnet Data Type*
ANALOG INPUT		
AI	AckedTransitions	BACnetEventTransitionBits
AI	CovIncrement	Real
AI	Deadband	Real
AI	Description	CharacterString
AI	DeviceType	CharacterString
AI	EventAlgorithmInhibit	Boolean
AI	EventDetectionEnable	Boolean
AI	EventEnable	BACnetEventTransitionBits
AI	EventMessageTexts	CharacterString
AI	EventMessageTextsConfig	CharacterString
AI	EventState	BACnetEventState
AI	EventTimeStamps	BACnetTimeStamp
AI	FaultHighLimit	Real
AI	FaultLowLimit	Real

Object	*Property Name*	*BACnet Data Type*
AI	HighLimit	Real
AI	InterfaceValue	BACnetOptionalREAL
AI	LimitEnable	BACnetLimitEnable
AI	LowLimit	Real
AI	MaxPresValue	Real
AI	MinPresValue	Real
AI	NotificationClass	Unsigned
AI	NotifyType	BACnetNotifyType
AI	ObjectIdentifier	BACnetObjectIdentifier
AI	ObjectName	CharacterString
AI	ObjectType	BACnetObjectType
AI	OutOfService	Boolean
AI	PresentValue	Real
AI	ProfileLocation	CharacterString
AI	ProfileName	CharacterString
AI	PropertyList	BACnetPropertyIdentifier
AI	ReliabilityEvaluationInhibit	Boolean
AI	Resolution	Real
AI	StatusFlags	BACnetStatusFlags
AI	Tags	BACnetNameValue
AI	TimeDelay	Unsigned
AI	TimeDelayNormal	Unsigned
AI	Units	BACnetEngineeringUnits
AI	UpdateInterval	Unsigned
ANALOG OUTPUT		
AO	AckedTransitions	BACnetEventTransitionBits
AO	CommandTimeArray	BACnetTimeStamp
AO	CovIncrement	Real
AO	CurrentCommandPriority	BACnetOptionalUnsigned
AO	Deadband	Real
AO	Description	CharacterString
AO	DeviceType	CharacterString
AO	EventAlgorithmInhibit	Boolean
AO	EventDetectionEnable	Boolean
AO	EventEnable	BACnetEventTransitionBits
AO	EventMessageTexts	CharacterString
AO	EventMessageTextsConfig	CharacterString
AO	EventState	BACnetEventState
AO	EventTimeStamps	BACnetTimeStamp
AO	HighLimit	Real
AO	InterfaceValue	BACnetOptionalREAL
AO	LastCommandTime	BACnetTimeStamp
AO	LimitEnable	BACnetLimitEnable
AO	LowLimit	Real
AO	MaxPresValue	Real
AO	MinPresValue	Real

Object	*Property Name*	*BACnet Data Type*
AO	NotificationClass	Unsigned
AO	NotifyType	BACnetNotifyType
AO	ObjectIdentifier	BACnetObjectIdentifier
AO	ObjectName	CharacterString
AO	ObjectType	BACnetObjectType
AO	OutOfService	Boolean
AO	PresentValue	Real
AO	ProfileLocation	CharacterString
AO	ProfileName	CharacterString
AO	PropertyList	BACnetPropertyIdentifier
AO	Reliability	BACnetReliability
AO	ReliabilityEvaluationInhibit	Boolean
AO	RelinquishDefault	Real
AO	Resolution	Real
AO	StatusFlags	BACnetStatusFlags
AO	Tags	BACnetNameValue
AO	TimeDelay	Unsigned
AO	TimeDelayNormal	Unsigned
AO	Units	BACnetEngineeringUnits
ANALOG VALUE		
AV	AckedTransitions	BACnetEventTransitionBits
AV	CommandTimeArray	BACnetTimeStamp
AV	CovIncrement	Real
AV	CurrentCommandPriority	BACnetOptionalUnsigned
AV	Deadband	Real
AV	Description	CharacterString
AV	EventAlgorithmInhibit	Boolean
AV	EventDetectionEnable	Boolean
AV	EventEnable	BACnetEventTransitionBits
AV	EventMessageTexts	CharacterString
AV	EventMessageTextsConfig	CharacterString
AV	EventState	BACnetEventState
AV	EventTimeStamps	BACnetTimeStamp
AV	FaultHighLimit	Real
AV	FaultLowLimit	Real
AV	HighLimit	Real
AV	LastCommandTime	BACnetTimeStamp
AV	LimitEnable	BACnetLimitEnable
AV	LowLimit	Real
AV	MaxPresValue	Real
AV	MinPresValue	Real
AV	NotificationClass	Unsigned
AV	NotifyType	BACnetNotifyType
AV	ObjectIdentifier	BACnetObjectIdentifier
AV	ObjectName	CharacterString
AV	ObjectType	BACnetObjectType

Object	*Property Name*	*BACnet Data Type*
AV	OutOfService	Boolean
AV	PresentValue	Real
AV	ProfileLocation	CharacterString
AV	ProfileName	CharacterString
AV	PropertyList	BACnetPropertyIdentifier
AV	Reliability	BACnetReliability
AV	ReliabilityEvaluationInhibit	Boolean
AV	RelinquishDefault	Real
AV	Resolution	Real
AV	StatusFlags	BACnetStatusFlags
AV	Tags	BACnetNameValue
AV	TimeDelay	Unsigned
AV	TimeDelayNormal	Unsigned
AV	Units	BACnetEngineeringUnits
BINARY INPUT		
BI	AckedTransitions	BACnetEventTransitionBits
BI	ActiveText	CharacterString
BI	AlarmValue	BACnetBinaryPV
BI	ChangeOfStateCount	Unsigned
BI	ChangeOfStateTime	BACnetDateTime
BI	Description	CharacterString
BI	DeviceType	CharacterString
BI	ElapsedActiveTime	Unsigned32
BI	EventAlgorithmInhibit	Boolean
BI	EventDetectionEnable	Boolean
BI	EventEnable	BACnetEventTransitionBits
BI	EventMessageTexts	CharacterString
BI	EventMessageTextsConfig	CharacterString
BI	EventState	BACnetEventState
BI	EventTimeStamps	BACnetTimeStamp
BI	InactiveText	CharacterString
BI	InterfaceValue	BACnetOptionalBinaryPV
BI	NotificationClass	Unsigned
BI	NotifyType	BACnetNotifyType
BI	ObjectIdentifier	BACnetObjectIdentifier
BI	ObjectName	CharacterString
BI	ObjectType	BACnetObjectType
BI	OutOfService	Boolean
BI	Polarity	BACnetPolarity
BI	PresentValue	BACnetBinaryPV
BI	ProfileLocation	CharacterString
BI	ProfileName	CharacterString
BI	PropertyList	BACnetPropertyIdentifier
BI	Reliability	BACnetReliability
BI	ReliabilityEvaluationInhibit	Boolean
BI	StatusFlags	BACnetStatusFlags

Object	*Property Name*	*BACnet Data Type*
BI	Tags	BACnetNameValue
BI	TimeDelay	Unsigned
BI	TimeDelayNormal	Unsigned
BI	TimeOfActiveTimeReset	BACnetDateTime
BI	TimeOfStateCountReset	BACnetDateTime
BINARY OUTPUT		
BO	AckedTransitions	BACnetEventTransitionBits
BO	ActiveText	CharacterString
BO	ChangeOfStateCount	Unsigned
BO	ChangeOfStateTime	BACnetDateTime
BO	CommandTimeArray	BACnetTimeStamp
BO	CurrentCommandPriority	BACnetOptionalUnsigned
BO	Description	CharacterString
BO	DeviceType	CharacterString
BO	ElapsedActiveTime	Unsigned32
BO	EventAlgorithmInhibit	Boolean
BO	EventDetectionEnable	Boolean
BO	EventEnable	BACnetEventTransitionBits
BO	EventMessageTexts	CharacterString
BO	EventMessageTextsConfig	CharacterString
BO	EventState	BACnetEventState
BO	EventTimeStamps	BACnetTimeStamp
BO	FeedbackValue	BACnetBinaryPV
BO	InactiveText	CharacterString
BO	InterfaceValue	BACnetOptionalBinaryPV
BO	LastCommandTime	BACnetTimeStamp
BO	MinimumOffTime	Unsigned32
BO	MinimumOnTime	Unsigned32
BO	NotificationClass	Unsigned
BO	NotifyType	BACnetNotifyType
BO	ObjectIdentifier	BACnetObjectIdentifier
BO	ObjectName	CharacterString
BO	ObjectType	BACnetObjectType
BO	OutOfService	Boolean
BO	Polarity	BACnetPolarity
BO	PresentValue	BACnetBinaryPV
BO	ProfileLocation	CharacterString
BO	ProfileName	CharacterString
BO	PropertyList	BACnetPropertyIdentifier
BO	Reliability	BACnetReliability
BO	ReliabilityEvaluationInhibit	Boolean
BO	RelinquishDefault	BACnetBinaryPV
BO	StatusFlags	BACnetStatusFlags
BO	Tags	BACnetNameValue
BO	TimeDelay	Unsigned
BO	TimeDelayNormal	Unsigned

Object	*Property Name*	*BACnet Data Type*
BO	TimeOfActiveTimeReset	BACnetDateTime
BO	TimeOfStateCountReset	BACnetDateTime
BINARY VALUE		
BV	AckedTransitions	BACnetEventTransitionBits
BV	ActiveText	CharacterString
BV	AlarmValue	BACnetBinaryPV
BV	ChangeOfStateCount	Unsigned
BV	ChangeOfStateTime	BACnetDateTime
BV	CommandTimeArray	BACnetTimeStamp
BV	CurrentCommandPriority	BACnetOptionalUnsigned
BV	Description	CharacterString
BV	ElapsedActiveTime	Unsigned32
BV	EventAlgorithmInhibit	Boolean
BV	EventDetectionEnable	Boolean
BV	EventEnable	BACnetEventTransitionBits
BV	EventMessageTexts	CharacterString
BV	EventMessageTextsConfig	CharacterString
BV	EventState	BACnetEventState
BV	EventTimeStamps	BACnetTimeStamp
BV	InactiveText	CharacterString
BV	LastCommandTime	BACnetTimeStamp
BV	MinimumOffTime	Unsigned32
BV	MinimumOnTime	Unsigned32
BV	NotificationClass	Unsigned
BV	NotifyType	BACnetNotifyType
BV	ObjectIdentifier	BACnetObjectIdentifier
BV	ObjectName	CharacterString
BV	ObjectType	BACnetObjectType
BV	OutOfService	Boolean
BV	PresentValue	BACnetBinaryPV
BV	ProfileLocation	CharacterString
BV	ProfileName	CharacterString
BV	PropertyList	BACnetPropertyIdentifier
BV	Reliability	BACnetReliability
BV	ReliabilityEvaluationInhibit	Boolean
BV	RelinquishDefault	BACnetBinaryPV
BV	StatusFlags	BACnetStatusFlags
BV	Tags	BACnetNameValue
BV	TimeDelay	Unsigned
BV	TimeDelayNormal	Unsigned
BV	TimeOfActiveTimeReset	BACnetDateTime
BV	TimeOfStateCountReset	BACnetDateTime
DEVICE		
DV	AckedTransitions	BACnetEventTransitionBits

Object	*Property Name*	*BACnet Data Type*
DV	AlignIntervals	Boolean
DV	ApduSegmentTimeout	Unsigned
DV	ApduTimeout	Unsigned
DV	ApplicationSoftwareVersion	CharacterString
DV	BackupAndRestoreState	BACnetBackupState
DV	BackupFailureTimeout	Unsigned16
DV	BackupPreparationTime	Unsigned16
DV	ConfigurationFiles	BACnetObjectIdentifier
DV	DatabaseRevision	Unsigned
DV	DaylightSavingsStatus	Boolean
DV	DeployedProfileLocation	CharacterString
DV	Description	CharacterString
DV	EventDetectionEnable	Boolean
DV	EventEnable	BACnetEventTransitionBits
DV	EventMessageTexts	CharacterString
DV	EventMessageTextsConfig	CharacterString
DV	EventState	BACnetEventState
DV	EventTimeStamps	BACnetTimeStamp
DV	FirmwareRevision	CharacterString
DV	IntervalOffset	Unsigned
DV	LastRestartReason	BACnetRestartReason
DV	LastRestoreTime	BACnetTimeStamp
DV	LocalDate	Date
DV	LocalTime	Time
DV	Location	CharacterString
DV	MaxApduLengthAccepted	Unsigned
DV	MaxInfoFrames	Unsigned
DV	MaxMaster	Unsigned
DV	MaxSegmentsAccepted	Unsigned
DV	ModelName	CharacterString
DV	NotificationClass	Unsigned
DV	NotifyType	BACnetNotifyType
DV	NumberOfApduRetries	Unsigned
DV	ObjectIdentifier	BACnetObjectIdentifier
DV	ObjectList	BACnetObjectIdentifier
DV	ObjectName	CharacterString
DV	ObjectType	BACnetObjectType
DV	ProfileLocation	CharacterString
DV	ProfileName	CharacterString
DV	PropertyList	BACnetPropertyIdentifier
DV	ProtocolObjectTypesSupported	BACnetObjectTypesSupported
DV	ProtocolRevision	Unsigned
DV	ProtocolServicesSupported	BACnetServicesSupported
DV	ProtocolVersion	Unsigned
DV	Reliability	BACnetReliability
DV	ReliabilityEvaluationInhibit	Boolean
DV	RestoreCompletionTime	Unsigned16

Object	*Property Name*	*BACnet Data Type*
DV	RestorePreparationTime	Unsigned16
DV	SegmentationSupported	BACnetSegmentation
DV	SerialNumber	CharacterString
DV	StatusFlags	BACnetStatusFlags
DV	StructuredObjectList	BACnetObjectIdentifier
DV	SystemStatus	BACnetDeviceStatus
DV	Tags	BACnetNameValue
DV	TimeOfDeviceRestart	BACnetTimeStamp
DV	TimeSynchronizationInterval	Unsigned
DV	UtcOffset	Integer
DV	VendorIdentifier	Unsigned16
DV	VendorName	CharacterString
DV	VtClassesSupported	BACnetVTClass
GROUP		
GR	Description	CharacterString
GR	ObjectIdentifier	BACnetObjectIdentifier
GR	ObjectName	CharacterString
GR	ObjectType	BACnetObjectType
GR	PresentValue	ReadAccessResult
GR	ProfileLocation	CharacterString
GR	ProfileName	CharacterString
GR	PropertyList	BACnetPropertyIdentifier
GR	Tags	BACnetNameValue
MULTI-STATE INPUT		
MI	AckedTransitions	BACnetEventTransitionBits
MI	AlarmValues	Unsigned
MI	Description	CharacterString
MI	DeviceType	CharacterString
MI	EventAlgorithmInhibit	Boolean
MI	EventDetectionEnable	Boolean
MI	EventEnable	BACnetEventTransitionBits
MI	EventMessageTexts	CharacterString
MI	EventMessageTextsConfig	CharacterString
MI	EventState	BACnetEventState
MI	EventTimeStamps	BACnetTimeStamp
MI	FaultValues	Unsigned
MI	InterfaceValue	BACnetOptionalUnsigned
MI	NotificationClass	Unsigned
MI	NotifyType	BACnetNotifyType
MI	NumberOfStates	Unsigned
MI	ObjectIdentifier	BACnetObjectIdentifier
MI	ObjectName	CharacterString
MI	ObjectType	BACnetObjectType
MI	OutOfService	Boolean

Object	*Property Name*	*BACnet Data Type*
MI	PresentValue	Unsigned
MI	ProfileLocation	CharacterString
MI	ProfileName	CharacterString
MI	PropertyList	BACnetPropertyIdentifier
MI	Reliability	BACnetReliability
MI	ReliabilityEvaluationInhibit	Boolean
MI	StateText	CharacterString
MI	StatusFlags	BACnetStatusFlags
MI	Tags	BACnetNameValue
MI	TimeDelay	Unsigned
MI	TimeDelayNormal	Unsigned
MULTI-STATE OUTPUT		
MO	AckedTransitions	BACnetEventTransitionBits
MO	CommandTimeArray	BACnetTimeStamp
MO	CurrentCommandPriority	BACnetOptionalUnsigned
MO	Description	CharacterString
MO	DeviceType	CharacterString
MO	EventAlgorithmInhibit	Boolean
MO	EventDetectionEnable	Boolean
MO	EventEnable	BACnetEventTransitionBits
MO	EventMessageTexts	CharacterString
MO	EventMessageTextsConfig	CharacterString
MO	EventState	BACnetEventState
MO	EventTimeStamps	BACnetTimeStamp
MO	FeedbackValue	Unsigned
MO	InterfaceValue	BACnetOptionalUnsigned
MO	LastCommandTime	BACnetTimeStamp
MO	NotificationClass	Unsigned
MO	NotifyType	BACnetNotifyType
MO	NumberOfStates	Unsigned
MO	ObjectIdentifier	BACnetObjectIdentifier
MO	ObjectName	CharacterString
MO	ObjectType	BACnetObjectType
MO	OutOfService	Boolean
MO	PresentValue	Unsigned
MO	ProfileLocation	CharacterString
MO	ProfileName	CharacterString
MO	PropertyList	BACnetPropertyIdentifier
MO	Reliability	BACnetReliability
MO	ReliabilityEvaluationInhibit	Boolean
MO	RelinquishDefault	Unsigned
MO	StateText	CharacterString
MO	StatusFlags	BACnetStatusFlags
MO	Tags	BACnetNameValue
MO	TimeDelay	Unsigned
MO	TimeDelayNormal	Unsigned

Object	*Property Name*	*BACnet Data Type*
MULTI-STATE VALUE		
MV	AckedTransitions	BACnetEventTransitionBits
MV	AlarmValues	Unsigned
MV	CommandTimeArray	BACnetTimeStamp
MV	CurrentCommandPriority	BACnetOptionalUnsigned
MV	Description	CharacterString
MV	EventAlgorithmInhibit	Boolean
MV	EventDetectionEnable	Boolean
MV	EventEnable	BACnetEventTransitionBits
MV	EventMessageTexts	CharacterString
MV	EventMessageTextsConfig	CharacterString
MV	EventState	BACnetEventState
MV	EventTimeStamps	BACnetTimeStamp
MV	FaultValues	Unsigned
MV	LastCommandTime	BACnetTimeStamp
MV	NotificationClass	Unsigned
MV	NotifyType	BACnetNotifyType
MV	NumberOfStates	Unsigned
MV	ObjectIdentifier	BACnetObjectIdentifier
MV	ObjectName	CharacterString
MV	ObjectType	BACnetObjectType
MV	OutOfService	Boolean
MV	PresentValue	Unsigned
MV	ProfileLocation	CharacterString
MV	ProfileName	CharacterString
MV	PropertyList	BACnetPropertyIdentifier
MV	Reliability	BACnetReliability
MV	ReliabilityEvaluationInhibit	Boolean
MV	RelinquishDefault	Unsigned
MV	StateText	CharacterString
MV	StatusFlags	BACnetStatusFlags
MV	Tags	BACnetNameValue
MV	TimeDelay	Unsigned
MV	TimeDelayNormal	Unsigned
LIFE SAFETY POINT		
LSP	AcceptedModes	BACnetLifeSafetyMode
LSP	AckedTransitions	BACnetEventTransitionBits
LSP	AlarmValues	BACnetLifeSafetyState
LSP	Description	CharacterString
LSP	DeviceType	CharacterString
LSP	DirectReading	Real
LSP	EventAlgorithmInhibit	Boolean
LSP	EventDetectionEnable	Boolean
LSP	EventEnable	BACnetEventTransitionBits

Object	*Property Name*	*BACnet Data Type*
LSP	EventMessageTexts	CharacterString
LSP	EventMessageTextsConfig	CharacterString
LSP	EventState	BACnetEventState
LSP	EventTimeStamps	BACnetTimeStamp
LSP	FaultValues	BACnetLifeSafetyState
LSP	LifeSafetyAlarmValues	BACnetLifeSafetyState
LSP	MaintenanceRequired	BACnetMaintenance
LSP	MemberOf	BACnetDeviceObjectReference
LSP	Mode	BACnetLifeSafetyMode
LSP	NotificationClass	Unsigned
LSP	NotifyType	BACnetNotifyType
LSP	ObjectIdentifier	BACnetObjectIdentifier
LSP	ObjectName	CharacterString
LSP	ObjectType	BACnetObjectType
LSP	OperationExpected	BACnetLifeSafetyOperation
LSP	OutOfService	Boolean
LSP	PresentValue	BACnetLifeSafetyState
LSP	ProfileLocation	CharacterString
LSP	ProfileName	CharacterString
LSP	PropertyList	BACnetPropertyIdentifier
LSP	Reliability	BACnetReliability
LSP	ReliabilityEvaluationInhibit	Boolean
LSP	Setting	Unsigned8
LSP	Silenced	BACnetSilencedState
LSP	StatusFlags	BACnetStatusFlags
LSP	Tags	BACnetNameValue
LSP	TimeDelay	Unsigned
LSP	TimeDelayNormal	Unsigned
LSP	TrackingValue	BACnetLifeSafetyState
LSP	Units	BACnetEngineeringUnits

Table 9.2 – Supported BACnet Object Properties

10.TECHNICAL SPECIFICATIONS

10.1. DIMENSIONS

Below are the enclosure dimensions as well as the required DIN rail dimensions. All dimensions are in millimeters.

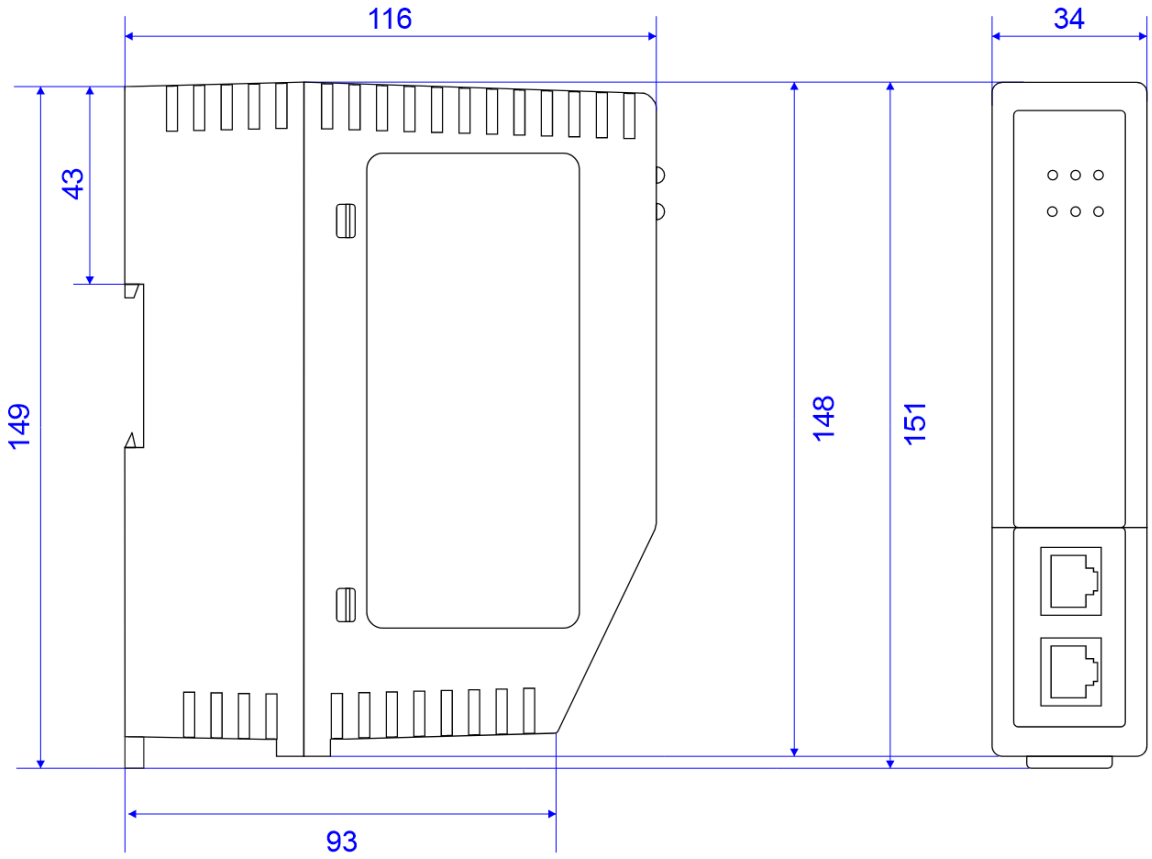


Figure 10.1 – BACnet Master enclosure dimensions

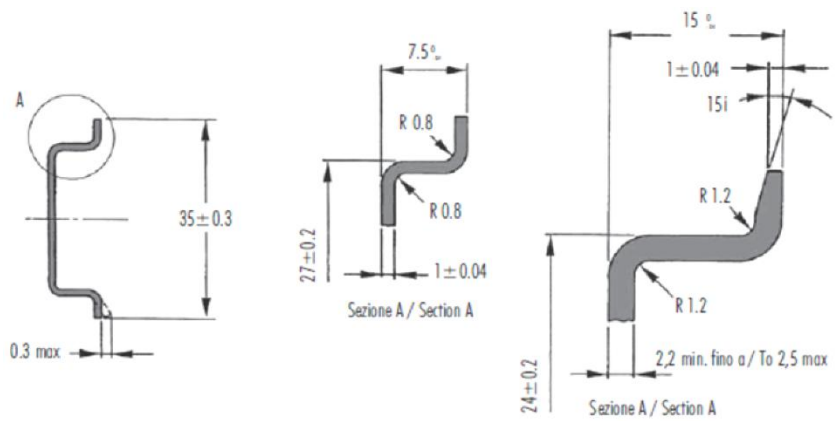


Figure 10.2 - Required DIN dimensions

10.1. ELECTRICAL

Specification	Rating
Power requirements	Input: 10 – 32V DC, (121 mA @ 24 VDC)
Voltage Fluctuations	Voltage fluctuations < ±10% Transient Over-voltages up to the levels of OVERVOLTAGE CATEGORY I
Power consumption	3.1 W (Including full load on USB of 200mA) 300 mA maximum
USB Power	5V, maximum load of 200 mA (1W).
Connector	3-way terminal
Conductors	24 – 18 AWG
Earth connection	Yes, terminal based
Emissions	IEC61000-6-4
ESD Immunity	EN 61000-4-2
Radiated RF Immunity	IEC 61000-4-3
EFT/B Immunity	EFT: IEC 61000-4-4
Surge Immunity	Surge: IEC 61000-4-5
Conducted RF Immunity	IEC 61000-4-6

Table 10.1 - Electrical specification.

10.2. ENVIRONMENTAL

Specification	Rating
Enclosure rating	IP20, NEMA/UL Open Type Indoor use only
Temperature	-20 – 70 °C
Relative Humidity	5% to 90% - No condensation
Pollution Degree	2
Altitude	< 2000 m

Table 10.2 - Environmental specification.

10.3. ETHERNET

Specification	Rating
Connector	RJ45
Conductors	CAT5 STP/UTP
ARP connections	Max 200
TCP connections	Max 40
CIP connections	Max 20
Communication rate	10/100Mbps
Duplex mode	Full/Half
Auto-MDIX support	Yes
Controller Support	ControlLogix, CompactLogix
Embedded switch	Yes, 2 x Ethernet ports
Device Level Ring (DLR)	Supported
Network Time Protocol (NTP)	Supported

Table 10.3 - Ethernet specification.

10.4. SERIAL PORT (RS232)

Specification	Rating
RS232 Connector	9-way terminal (shared with RS485)
RS232 Conductor	24 – 18 AWG
Electrical Isolation	1000 Vdc
BAUD	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Parity	None, Even, Odd
Data bits	8
Stop bits	1

Table 10.4 – RS232 Serial Port specification.

10.5. SERIAL PORT (RS485)

Specification	Rating
RS485 Connector	9-way terminal (shared with RS232)
RS485 Conductor	24 – 18 AWG
Electrical Isolation	1500 Vrms for 1 minute.
BAUD	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Parity	None, Even, Odd
Data bits	8
Stop bits	1

Table 10.5 – RS485 Serial Port specification.

10.6. BACNET

Specification	Rating
Maximum BACnet Points	200 (A-BNM-B) 1000 (A-BNM-P) 2500 (A-BNM-M)
Maximum BACnet Devices	50 (A-BNM-B) 250 (A-BNM-P) 250 (A-BNM-M)
Mapped Properties per Device	256
BACnet Protocols Supported	BACnet MSTP (485) BACnet /IP BACnet /Ethernet
Supported Functions	Read, Write, Write-on-change, COV (change-of-value), COV Property, COV Property Increment.
Supported Objects	See section 9.1.

Table 10.6 – BACnet specification

10.7. ETHERNET/IP TARGET

Specification	Rating
Class 1 Cyclic connection count	25
Minimum RPI	20 ms (1 - 5 total connections) 40 ms (6 - 10 total connections) 60 ms (11 - 15 total connections) 80 ms (16 - 20 total connections) 100 ms (21 - 25 total connections)

Table 10.7 – EtherNet/IP Target specification

10.8. ETHERNET/IP ORIGINATOR

Specification	Rating
Class 1 Cyclic Connections Supported	Yes
Class 3 / UCMM Connections Supported	Yes
Class 1 Connection Count	10
Class 3 / UCMM Target Device Count	10
Class 3 / UCMM Mapping Count	50
Logix Direct-to-Tag Supported	Yes

Table 10.8 – EtherNet/IP Originator specification

10.9. MODBUS CLIENT

Specification	Rating
Modes Supported	Modbus TCP, Modbus RTU232, Modbus RTU485
Max. Modbus Server Devices	20
Max. Modbus Mapping	100
Mapping Ranges	Holding Register 0 – 4999 Input Register 0 – 4999 Input Status 0 – 4999 Coil Status 0 – 4999
Base Offset	Modbus (Base 0)

	PLC (Base 1)
Configurable Modbus TCP Port	Yes
Data Reformatting Supported	BB AA BB AA DD CC CC DD AA BB DD CC BB AA

Table 10.9 – Modbus Client specification

10.10. MODBUS SERVER

Specification	Rating
Modes Supported	Modbus TCP, Modbus RTU232, Modbus RTU485
Mapping Ranges	Holding Register 0 – 4999 Input Register 0 – 4999 Input Status 0 – 4999 Coil Status 0 – 4999
Base Offset	Modbus (Base 0) PLC (Base 1)
Configurable Modbus TCP Port	Yes

Table 10.10 – Modbus Server specification

10.11. CERTIFICATIONS

Certification	Mark
CE Mark	CE
RoHS2 Compliant	RoHS2
UL Mark File: E494895	c UL us LISTED CLASS 1, DIV 2, GROUPS A, B, C, D T5

UKCA	
ODVA Conformance	

Table 10.11 – Certifications

11.INDEX

A

Adanced, 60, 61

B

BACNET Communication configuration, 31
BACNET general configuration, 28
BACnet Master, 15, 16, 27, 135, 155
BACnet Master, 8
BACnet Master, 93
BACnet Master, 144
BACnet Master, 155
BACnet Master, 165
BACnet Master general configuration, 43, 44, 46, 48, 63, 82
BACNET Slave configuration, 33

C

Connection path, 93
Contact Us, 14
ControlLogix, 8

D

DC power, 15
DHCP, 16, 21, 22, 23, 24
dimensions, 187
DIN rail, 17, 187
DIP, 16

E

Ethernet connector, 19
Ethernet TCP, 8
Ethernet/IP, 61
EtherNet/IP, 8, 61
EtherNet/IP Devices configuration, 60
EtherNet/IP Map configuration, 61, 90, 91, 92

F

firmware upgrade, 28

G

General parameters, 28

I

input voltage, 18
IP Address, 22, 23

L

LED, 75, 143, 144
Logix platform, 8

M

MODBUS, 156
Modbus IPBlock, 8

O

output assembly, 135, 140

R

Rockwell Automation, 26
RS232, 15, 18, 19
RSLinx, 26

S

Safe Mode, 16
Serial, 189, 190
Slate, 28, 94, 144, 159, 165
SLATE Configuration Utility, 106, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128
statistics, 153, 154, 155, 158, 161, 167, 171
Statistics, 144
Support email, 14

T

Target Browser, 24, 25, 61, 93

W

web server, 144