



User Guide

Babel Buster 3 BACnet Router

Model BB3-7302-V2C
Model MX-73-V2C
Rev. 1.1 – June 2021

© 2021 Control Solutions, Inc.

BB3-7302/MX-73-V2C User Guide Contents

1 Introduction

- 1.1 How to Use This Guide
- 1.2 Important Safety Notice
- 1.3 Warranty

2 Connecting Gateway for the First Time

- 2.1 Where to Start
- 2.2 Overview of Model BB3-7302/MX-73-V2C
 - 2.2.1 Application of the BB3-7302/MX-73-V2C
 - 2.2.2 Object Server Model for a Gateway
- 2.3 What is New in Model BB3-7302/MX-73-V2C
- 2.4 Connectors and Indicators
- 2.5 Opening the Web User Interface

3 System Configuration and Resources

- 3.1 Using the File Manager
 - 3.1.1 Load, Save, Create XML Configuration File
 - 3.1.2 Select Startup Configuration
 - 3.1.3 Delete a File
 - 3.1.4 Import CSV File
 - 3.1.5 Clear Configuration
- 3.2 Configuration Files and Restoring Default Settings
- 3.3 Network Configuration
 - 3.3.1 IPv4 Settings
 - 3.3.2 IPv6 Settings
 - 3.3.3 NTP Time Server Settings
 - 3.3.4 DNS Settings
 - 3.3.5 Setting Up Secure Web Server
 - 3.3.6 Port Settings
- 3.4 Resource Allocation
- 3.5 User Login Passwords

4 Configuring the BACnet Device

- 4.1 Device Object Parameters
- 4.2 Network Settings

5 Configuring BACnet Router

- 5.1 Router Settings
- 5.2 Configured Routes
- 5.3 Discovered Routes
- 5.4 Slave Proxy

6 Configuring Local Data Objects

- 6.1 Behavior of Input vs Output Objects
- 6.2 Allocating Local Objects
- 6.3 Configuring Local Objects
 - 6.3.1 Analog Input Objects
 - 6.3.2 Analog Output Objects
 - 6.3.3 Analog Value Objects
 - 6.3.4 Binary Input Objects
 - 6.3.5 Binary Output Objects
 - 6.3.6 Binary Value Objects
 - 6.3.7 Multistate Input Objects
 - 6.3.8 Multistate Output Objects
 - 6.3.9 Multistate Value Objects
- 6.4 Local Object Calculate Rules

- 6.5 Local Object Copy Rules
- 6.6 Device Status Reporting

[7 Configuring Gateway as a BACnet Client](#)

- 7.1 BACnet Device List
- 7.2 BACnet Client Read Maps
- 7.3 BACnet Client Write Maps
- 7.4 BACnet Client Diagnostics
- 7.5 Importing BACnet Client Maps from CSV File
- 7.6 Clearing Configuration
- 7.7 Understanding BACnet Client Timeout Settings

[8 Configuring Gateway as a BACnet Server](#)

- 8.1 BACnet Server Configuration
- 8.1 Accessing Local Objects

[9 Configuring BBMD](#)

- 9.1 Registering as a Foreign Device
- 9.2 Allowing Other Devices to Register Locally
- 9.3 Broadcast Distribution Table

[10 Configuring Gateway as an SNMP Server](#)

- 10.1 Creating Local SNMP MIB
- 10.2 Supported Data Formats, RFC 6340
- 10.3 Testing the SNMP Agent

[11 Configuring SNMP Trap Sender](#)

- 11.1 SNMP Trap Destinations
- 11.2 SNMP Trap Triggers
- 11.3 SNMP Trap Summary
- 11.4 SNMP Identity
- 11.5 Testing the SNMP Trap Sender

[12 Configuring Gateway as an SNMP Client](#)

- 12.1 SNMP Device Configuration
- 12.2 SNMP Client Read Maps (Get)
- 12.3 SNMP Client Write Maps (Set)
- 12.4 SNMP Client Data Displayed by Agent
- 12.5 Supported Data Formats
- 12.6 SNMP Errors

[13 Configuring SNMP Table Walker](#)

- 13.1 Table Walk Methods
- 13.2 Configuring Table Walk Rules
- 13.3 Table Walk Errors

[14 Configuring SNMP Trap Receiver](#)

- 14.1 Trap Receive Devices
- 14.2 Trap Receive Rules
- 14.3 Trap Receive Examples for SNMPv1
- 14.4 Trap Receive Example for SNMPv2

[Appendix A Hardware Details](#)

- A.1 Wiring
- A.2 Front Panel LED Indicators
- A.3 RS-485 Line Termination and Bias
- A.4 Soft Configuration Reset
- A.5 Discovering Lost IP Address
- A.6 Forced Hard Configuration Reset
- A.7 Firmware Update Notes

[Appendix B Trouble Shooting](#)

- B.1 File Upload Errors
- B.2 BACnet Trouble Shooting
 - B.2.1 General Trouble Shooting
 - B.2.2 Error Codes, Reliability Codes
 - B.2.3 Auto-Reset Errors
 - B.2.4 Trouble Shooting BACnet IP
 - B.2.5 Trouble Shooting BACnet MS/TP

- B.2.6 Using Wireshark
- B.2.7 Using Network Discovery Tool
- B.3 SNMP Trouble Shooting

[Appendix C CSV File Formats](#)

- C.1 BACnet Client Read/Write Maps
- C.2 BACnet Object Types

[Appendix D Object Properties](#)

- D.1 Data Object Properties (Analog, Binary, Multistate)
- D.2 Device Object Properties

[Appendix E BACnet Codes](#)

- E.1 BACnet Object Property Codes
- E.2 BACnet Engineering Units Codes

[Appendix F Using Wireshark for Trouble Shooting](#)

- F.1 Hardware Requirements for Ethernet
- F.2 Hardware Requirements for MS/TP
- F.3 Examples of Using Wireshark

[Appendix G SSL Certificates for Secure Web \(HTTPS\)](#)

- G.1 X.509 Auto-Certificate Generation
- G.2 External Certificates
- G.3 Certificate Generation Script (Linux)

[Appendix H Converting Older XML Files](#)

- H.1 BB2-7030 to BB3-7302/MX-73 Conversion
- H.2 CSV File Export



1. Introduction

1.1 How to Use This Guide

This user guide provides background information on how the gateway works, and an overview of the configuration process. There are several sections for groups of tabs found in the web interface in the gateway which is accessed by opening a web browser and browsing to the IP address of the device.

You should at least read Sections 2 and 3, and other sections specific to your intended use. There is a "Quick Help" section at the bottom of each web page in the gateway which is generally sufficient for quick reference in setting up the gateway.

1.2 Important Safety Notice

Proper system design is required for reliable and safe operation of distributed control systems incorporating any Control Solutions product. It is extremely important for the user and system designer to consider the effects of loss of power, loss of communications, and failure of components in the design of any monitoring or control application. This is especially important where the potential for property damage, personal injury, or loss of life may exist. By using ANY Control Solutions, Inc., product, the user has agreed to assume all risk and responsibility for proper system design as well as any consequence for improper system design.

1.3 Warranty

This documentation is provided "as is," without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Control Solutions may make improvements and/or changes in this documentation at any time. This documentation could include technical inaccuracies, typographical errors, and the like. Changes are periodically made to the information herein; these changes may be made without notice.

Product Warranty: All Control Solutions products are warranted against defects in materials and workmanship for a period of time from date of shipment from factory as follows: Two years on non-mechanical parts, one year on mechanical parts (e.g. relays). Defective units will be repaired or replaced, at manufacturer's discretion, at no cost to user except when negligence or improper use has resulted in damage. The express warranty stated herein is in lieu of all other warranties, express or implied,

including without limitation any warranties of merchantability or fitness for a particular purpose and all other warranties are hereby disclaimed and excluded by Control Solutions, Inc.

Configuration errors made by customer are not covered under warranty. Damage caused by incorrect electrical connection is not covered under warranty. Removing circuit boards from their enclosures will void the warranty - the complete product with all of its original circuit boards and components must be returned for warranty consideration.



2. Connecting Gateway for the First Time

2.1 Where to Start

The Babel Buster BB3-7302 can be used as a BACnet Router, or a gateway, or both. As a router, it routes BACnet IP traffic to the BACnet MS/TP network and vice versa. As a gateway, it can remap BACnet object data or share BACnet data with SNMP devices.

Start by getting familiar with this User Guide and the sections that pertain to your application. Be sure to review the remainder of this section. Online videos are also available to demonstrate key operations in setting up the BB3-7302/MX-73.

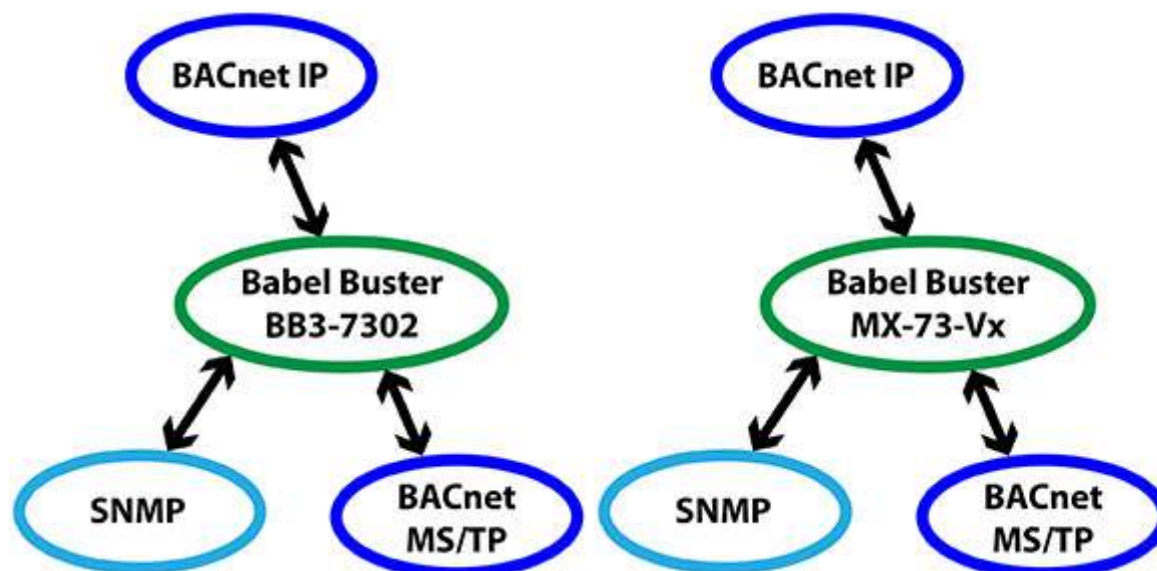
If you are going to be using the BB3-7302/MX-73 as a gateway to share data between SNMP and BACnet, you will need to obtain information such as the MIB from the vendor of the SNMP device you wish to connect. You will need to know what variables to read in order to interface an SNMP device to the BACnet network.

If you are going to be using the BB3-7302/MX-73 as a BACnet router, or as a gateway, you will need to determine the necessary network settings.

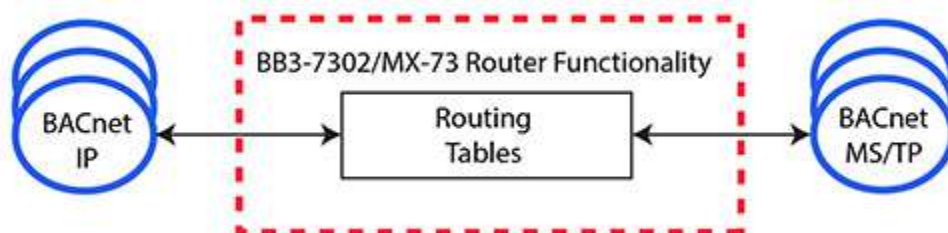
If you get stuck, you can open a support ticket at <https://ticket.csimn.com> where response time is generally 24 hours or less, and often as little as 2 hours, and at no cost.

2.2 Overview of Model BB3-7302/MX-73

2.2.1 Application of the BB3-7302/MX-73



The most common application of the BB3-7302/MX-73 will be to use it simply as a BACnet Router. In this case, you do not need to concern yourself with any of the client read/write maps. These are not necessary for routing. You simply need to set up the network parameters, and communication between BACnet IP and BACnet MS/TP devices will happen automatically.



When functioning as a Router, the BB3-7302/MX-73 will look at the address information contained in incoming messages, refer to its routing tables to look up network information, and forward the message with new address information that will take the message to its final destination, or to the next router.

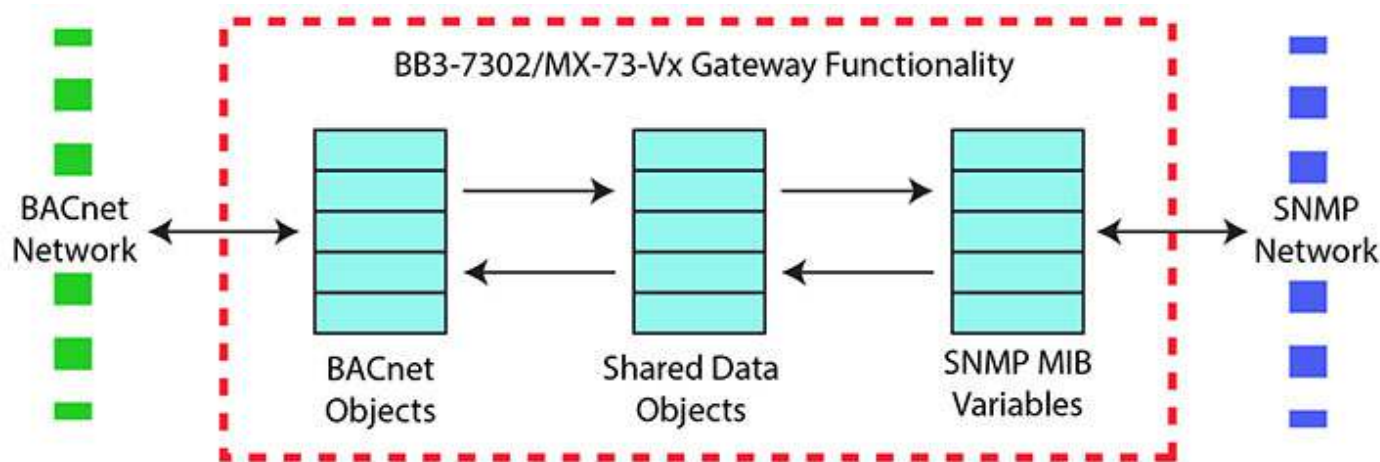
When functioning as a Gateway, the BB3-7302/MX-73 will exchange data between SNMP and BACnet devices. The BB3-7302/MX-73 can also be used as a gateway to exchange data between BACnet devices when there is a desire to simply remap objects. As a gateway, the BB3-7302/MX-73 may be thought of as a data server with multiple network ports that have access to the internal database. Both BACnet and SNMP have access to the internal data objects which are updated according to rules you define.

2.2.2 Object Server Model for a Gateway

Control Solutions gateways are not simple protocol translators. It is not possible to do an effective job of simply converting one protocol directly to another. Any attempt to do so would likely have negative effects on the networks on both sides of the gateway. An effective solution requires an intelligent device that can properly and efficiently act as a native device on each network. Control Solutions gateways function as two native

devices, one on each network, with a shared data base in between them. They function as clients and/or servers on each network.

The central data element in every Control Solutions gateway is an “object”. Each object has rules for accessing that object which are specific to the protocol of the network. Each object has at least two sets of rules, one set for each of the two (or more) networks that may access the object. The object model is often optimized to cater to a specific protocol, and will most often favor the more complex protocol.



Control Solutions gateways will function as servers, providing a copy of the most recent data found in its data base when a client requests that data. In master/slave terms, the server is a slave while the client is a master. Some applications will treat the gateway as a server from both (all) networks connected. But most applications will want the gateway to be a server on one side, and a client on the other side. The most frequent application of the BB3-7302/MX-73 as a gateway will have it functioning as an SNMP client.

Client functionality of a Control Solutions gateway is autonomous. In other words, when acting as an SNMP client, the gateway will continuously poll the SNMP server device(s) on its own, and keep a copy of the most recent data obtained from (or sent to) the SNMP server device(s). Most often, the gateway is configured to read server devices periodically, and write to the server devices when new data is received from a client.

It should be noted for clarity that no data is retained by the BB3-7302/MX-73 when functioning as a BACnet Router. The address header on the incoming packet is updated as applicable and immediately retransmitted on the other network without making any interpretation of the content of the packet.

2.3 What is New in Model BB3-7302/MX-73

The BB3-7302 is a significant enhancement over its predecessor, the BB2-7030. The hardware includes a faster processor and hardware encryption engine for efficient rendering of secure web pages. The software includes numerous enhancements.

- Secure (HTTPS) web server
- Faster processor

- Higher point count, up to 1,000 MIB variables
- User defined register map for Modbus
- Greater flexibility in assigning local register data types
- Support for reading character strings from Modbus
- CSV import of register maps for client/master configuration
- Menu options to clear part or all of configuration

IMPORTANT: Configuration files from older gateways are not directly usable in the BB3-7302/MX-73, but the Babel Buster Configuration Builder program can be used to convert a BB2-7030 configuration into CSV files that may be imported into the BB3-7302/MX-73.

2.4 Connectors and Indicators

Follow these steps to make the initial connection to the BB3-7302/MX-73.

(a) Connect power. Apply +12VDC to +24VDC or 24VAC to the terminal marked "POWER", and common or ground to one of the terminals marked "GND".



(b) Connect a CAT5 cable between the RJ-45 jack on the gateway, and your network switch or hub. You cannot connect directly to your PC unless you use a "crossover" cable (or your PC supports auto-MDIX, which many newer laptops do).

(c) Apply power.

A blue LED inside the case should light indicating power is present.

If the link LED on the RJ45 jack is not on, check your Ethernet cable connections. Both link and activity LEDs on the RJ45 jack will be on solid for a short time during boot-up. The entire bootup process will take 1-2 minutes, during which time you will not be able to connect with a browser.

Ethernet link LED is the yellow LED integrated into the CAT5 connector. Ethernet

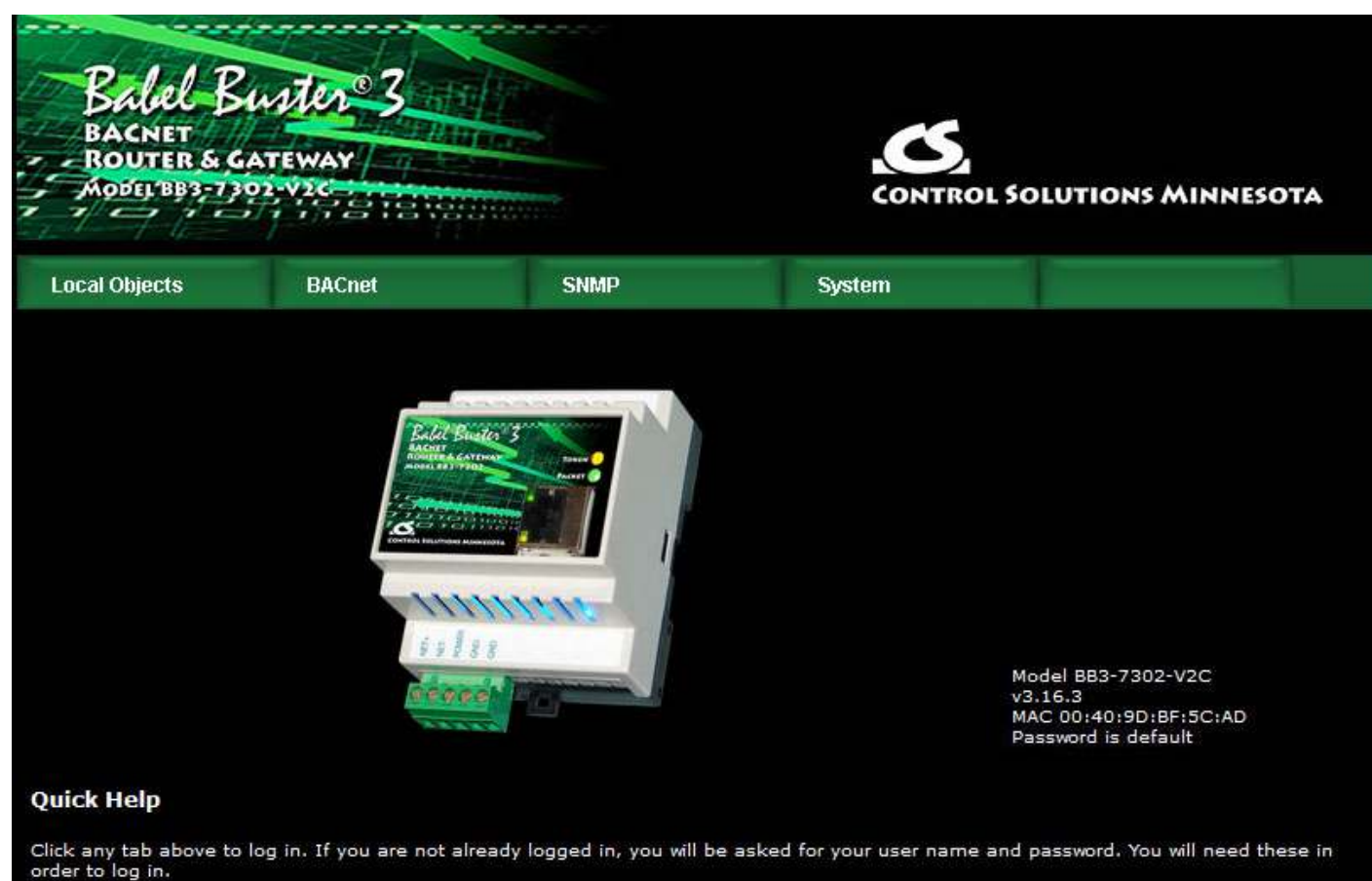
activity LED is the green LED integrated into the CAT5 connector.

Refer to Appendix A for additional detail pertaining to connections and indicators as well as optional internal jumper settings.

2.5 Opening the Web User Interface

The default IP address as shipped is 10.0.0.101. Open your browser, and enter "http://10.0.0.101/" in the address window. You should see a page with the "Babel Buster 3" header shown below (BB3-7302 is illustrated, MX-73 is similar). From this point, you will find help on each page in the web site contained within the product.

If your PC is not already on the 10.0.0.0 domain, and you are unable to connect, you may need to temporarily change your computer's IP address to a static IP address that starts with 10.0.0. and ends with anything but 101.



When you click on any of the page tabs such as System, you will be asked for a user name and password. The only login as shipped is user name "root" with a unique password generated specifically for your Babel Buster. Your password should be included on a document included with the gateway, or on a label attached to the gateway.

If the unique automatically generated password is currently in effect for user "root", it will be indicated by "Password is default" as shown in the above screen shot. If you have changed the root password to something of your own making, then this line is absent.

There is no way to get the BB3-7302/MX-73 to show you what the default root password is. If you have lost track of it, make a note of the MAC address, and open a support ticket at <https://ticket.csimn.com> to request the default root password (you will need to provide the MAC address in order to obtain the password).

To change the IP address of the gateway, go to the Network page under System :: System Setup. The following page should appear (only top portion illustrated here). Change the IP address, and subnet mask and gateway if applicable. Click Change IP to save the changes. The process of programming this into Flash takes around half a minute. The new IP address only takes effect following the next system restart or power cycle.

The screenshot displays the web interface for the Babel Buster 3 BACnet Router & Gateway, Model BB3-7302-V2C. The interface has a dark green header with the product name and logo. Below the header is a navigation bar with tabs: Local Objects, BACnet, SNMP, System, File Manager, Network (selected), Resources, and User. The main content area shows the 'IPv4 Settings' section with radio buttons for 'Automatic' (selected) and 'Static'. Below this are input fields for 'IPv4 Static IP Address' (192.168.1.78), 'IPv4 Static Subnet Mask' (255.255.255.0), and 'IPv4 Static Gateway' (192.168.1.1). To the right, the 'IPv4 Configured IP Address' is 192.168.1.78, the 'IPv4 Subnet Mask' is 255.255.255.0, and the 'IPv4 Gateway' is 192.168.1.1. An 'Apply' button is next to the configured IP address. Below the IPv4 settings is the 'IPv6 Settings' section with radio buttons for 'Disabled' (selected), 'Automatic', and 'Static'. The 'IPv6 Link-Local IP Address' is fe80::240:9dff:febf:5cad.

Most changes are stored in an XML configuration file in the device's Flash file system. Only a few are stored differently, and the IP address is one of those. Normally, clicking Update on any configuration page only stores that configuration information to a temporary RAM copy of the configuration file. To make your changes other than IP address permanent, you must select your file, select the Save XML Config File action, and then click Execute on the File Manager page. Refer to Section 3 for more about the File Manager.

Local Objects	BACnet	SNMP	System		
System Setup					
File Manager	Network	Resources	User		

Free space: 1.37 MB

File Directory:

Filtered by:

Seleted File: Action:

Boot configuration ☐ Confirm



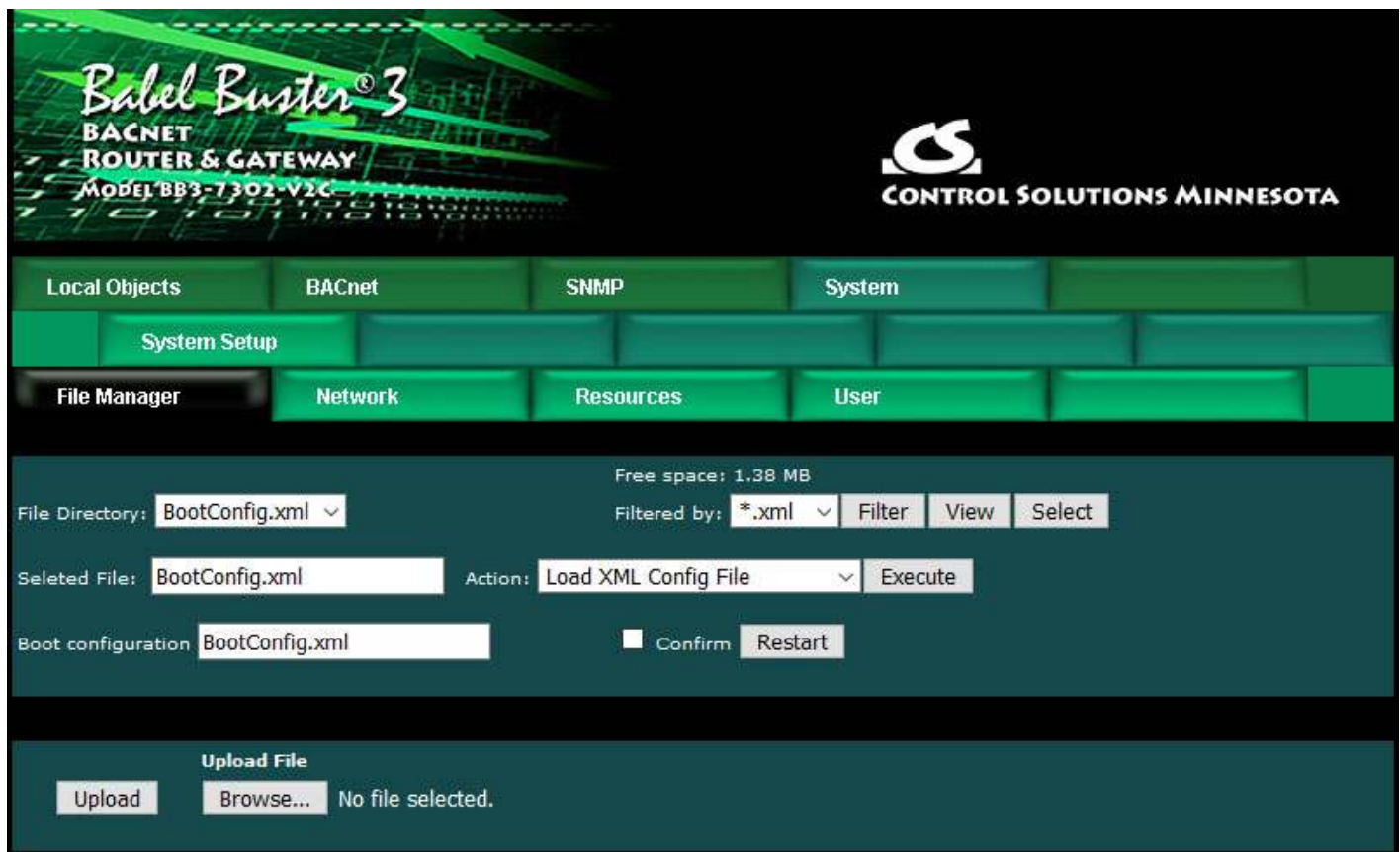
3. System Configuration and Resources

3.1 Using the File Manager

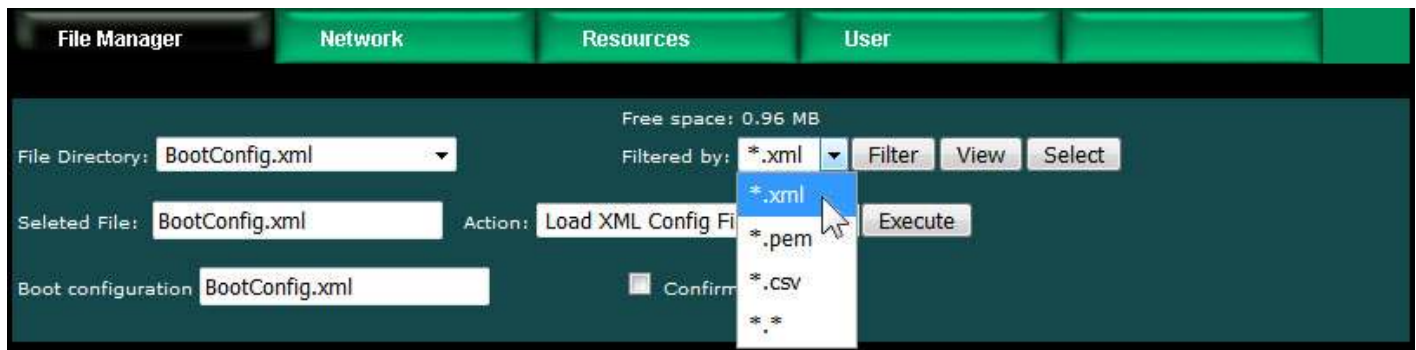
The File Manager page is probably one of the most important pages to know about. Among other things, this is where you tell the gateway to save all of the changes you have made. The various "Update" buttons on the many pages in the web user interface only copy your configuration from your PC's browser to temporary memory in the gateway. To retain those changes indefinitely (i.e. through restart or power cycle), you need to tell the gateway to save those changes in a configuration file.

The configuration files are stored in non-volatile (Flash) memory. The process of reprogramming the Flash takes a little time. It would be cumbersome to rewrite that file every time you made a minor change. Therefore, in the interest of being more responsive, and in the interest of extending the life of the Flash, configuration is only saved to Flash when you direct it to do so.

The File Manager is used in several other ways in addition to managing your XML configuration files. You upload SSL certificates here. You import CSV files for BACnet client configuration here.



The File Directory is a list of files that are currently stored in the Babel Buster's Flash file system. To filter files by type, select a type from the Filtered by list, and click Filter.

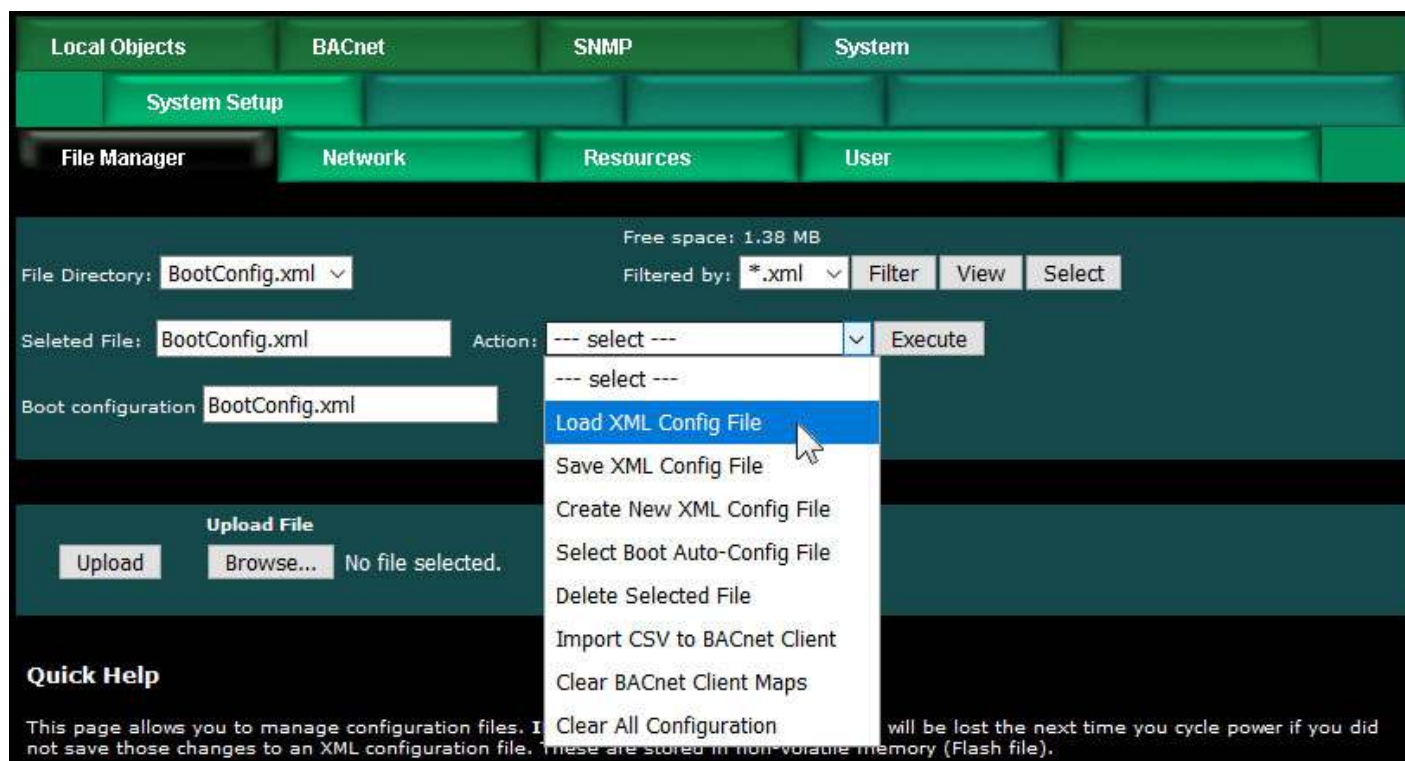


File type filters are as follows:

- *.xml XML configuration files
- *.pem SSL certificates (for AWS IoT and/or HTTPS)
- *.csv CSV spreadsheet for client map import
- *.* Display all files

There are several file related actions you may take. To take action with a certain file, select that file from the File Directory list, and click Select. That file should now show up in the Selected File window.

Once a file has been selected, choose your action from the Action list, and click Execute.



You must use the Select button to populate the Selected File window prior to executing any action from the list. Choose a file from the drop down list that shows all available files, then click the Select button. You may then act on that file.

You do not need to use the Select button to simply View a file. Clicking View will cause your browser to display the file chosen from the drop down list. If you attempt to View a CSV file, your PC will likely ask if you want to download the file or open it with your spread sheet program (e.g. Excel).

Upload File: To upload a file from your PC to this gateway, use the Browse button to find the file on your PC, open the file in the PC's file dialog box, and then click Upload.

NOTE: If you get a "File upload error: -1" message, click the browser's "back" button, then simply click the View button to view any file (does not matter which file), and then click browser's "back" button again to return to this page. This gets the browser and HTTP server back in sync, and this requirement generally happens only once following power-up.

Restart: To restart the gateway, check Confirm and click Restart. This is a hard reset that will accomplish the same thing as a power cycle without physically disconnecting and reconnecting power.

3.1.1 Load, Save, Create XML Configuration File

IMPORTANT: Configuration files from older gateways are not directly usable in the BB3-7302/MX-73, but the Babel Buster Configuration Builder program can be used to convert a BB2-7030 configuration into a BB3-7302/MX-73 configuration file. See Appendix H.

Load XML Config File: The configuration file shown in the "Boot configuration" window will be loaded automatically at startup. If you have uploaded a new configuration file and wish to use it without restarting, select that file (choose from list, click Select), select this action, and click Execute.

HINT: If you are loading a file generated externally and you get "parameter out of range" errors pertaining to defining objects or "table full" errors while loading maps or rules, you might not have sufficient resources allocated. You may need to increase some counts on the Resources page.

Save XML Config File: Any time you have made configuration changes that you want to retain as permanent, you need to come here, select the file from the directory list, and execute this Save action.

Create New XML Config File: You have the option to create a totally new configuration file. This is often suitable if you started with an existing configuration, made changes, and want to save your changes without replacing the original configuration. To create a new file, rather than selecting a file from the directory list, simply type a new name into the Selected file window. The name cannot contain spaces or special characters, and be sure to use the correct file suffix. Enter the name and execute this action.

3.1.2 Select Startup Configuration

Select Boot Auto-Config File: This is where you tell the Babel Buster what configuration to automatically load upon startup. To set the Boot configuration, select the XML file from the list, and execute this action. The name of the startup file, along with a few other important things like the gateway's own IP address, are stored in a different area of Flash that is not part of the file system.

When selecting a new Boot configuration file, it is a good idea to select the file, and execute Load XML Config File. If there are errors, they will be displayed. If there are errors in the file but you do not fix them, then the gateway will not fully start up the next time it restarts. The web user interface will be available, but it will not be talking to BACnet or SNMP devices.

3.1.3 Delete a File

Remove a file from the Flash file system by selecting it from the list and executing the Delete Selected File action.

3.1.4 Import CSV File

Import CSV to BACnet Client: You can configure BACnet client read and write maps in bulk by importing the maps as a CSV file that you created using a standard spreadsheet program. Refer to Appendix C for details about the CSV format. Note that maps will be added to the existing map list. If you want to replace existing maps with imported maps, execute Clear BACnet Client Maps first.

HINT: If you get "table full" errors while importing CSV files, you might not have sufficient resources allocated. You may need to increase some counts on the Resources page.

3.1.5 Clear Configuration

Clear BACnet Client Maps: Execute this action to clear (completely remove) all BACnet client read and write maps. The BACnet client device table will be left intact.

Clear All Configuration: Execute this action to completely wipe out all configuration. This includes the SNMP MIB definition, all SNMP client maps and devices, all BACnet client maps and devices, and all local objects. This will put you back to a "reset to factory" condition with the exception that your IP address is left unchanged. (See Appendix A, Section A.6, regarding forced hard configuration reset that includes IP address and root password.) If you want to make the now empty configuration permanent, select the file that is also selected as Boot configuration, and execute the Save XML Config File action.

The other means of completely wiping out all saved configuration is to simply delete the file named as the Boot configuration file, and then restart or power cycle the Babel Buster. Upon restart, a new empty configuration file will be created automatically.

3.2 Configuration Files and Restoring Default Settings

There is a means of restoring the Babel Buster to "manufacturer's default settings". First of all, make sure that the Boot configuration file is set to "BootConfig.xml". Then, after selecting this file as the boot file, delete it. Now restart the gateway. Upon restart, and upon finding that the boot configuration name is BootConfig.xml, and it does not exist, the gateway will automatically create one with default parameters. The automatic creation of a default file will not occur with any other file name.

Manual Editing: It is possible to manually edit the XML file outside of the gateway. However, doing so is very prone to errors. If there are errors in the XML file, it will not load successfully on startup. If the configuration does not load on startup, none of the scanners will begin scanning. Because they are all blocked by configuration failure, entering new configuration via the web pages will not result in functionality being restored. You must successfully load a configuration file before the gateway will become functional. To check for errors, select the file here, select Load XML Config File, and click Execute. Error messages that would have been discarded by the automatic loading at startup will now be displayed on an error page if there are any.

Backup Copy of XML Config File: To save a copy of the configuration to your PC, select the file and click the View button. Your browser will now display the XML file. DO NOT do a text copy/paste to try to create an XML file - doing so will result in an invalid file format that cannot be loaded again. You must use the browser's "save as" or "save page" function. The browser should default to wanting to save a file with a .xml suffix. If correctly saved on your PC, you should be able to double click on the saved file and it will result in opening the file automatically in your browser. It was saved correctly if the browser does not give any error messages when displaying

the XML (which should now look exactly as it did when you first clicked the View button). Saving the configuration file to your PC, and then uploading on a different device, is a quick and easy way to configure two Babel Busters the same way.

Note about caching: Your browser may cache files. If you view a file, make configuration changes, save the file, then view the file again, you may see the old file cached by the browser. To see the updated file, go to "Options" in your browser's tools menu, and delete temporary Internet files (or delete cache files). Also, if you upload a file, make changes on your PC, and re-upload the same file, the browser may send the old file. Again, you will need to find the button inside your browser options that lets you delete the cached files from your PC. To upload a configuration file from your PC to the gateway, use the Browse button to find the file on your PC, open the file in the PC's file dialog box, and then click Upload.

3.3 Network Configuration

The Network Configuration page is where you set the Babel Buster's IP address as well as a few other important things.

Babel Buster[®] 3
BACNET
ROUTER & GATEWAY
MODEL BB3-7302-V2C

CONTROL SOLUTIONS MINNESOTA

Local Objects BACnet SNMP System File Manager **Network** Resources User

IPv4 Settings ☒ Automatic ☐ Static

IPv4 Static IP Address: IPv4 Configured IP Address: **192.168.1.78**

IPv4 Static Subnet Mask: IPv4 Subnet Mask: **255.255.255.0**

IPv4 Static Gateway: IPv4 Gateway: **192.168.1.1**

IPv6 Settings ☒ Disabled ☐ Automatic ☐ Static

IPv6 Link-Local IP Address: **fe80::240:9dff:febf:5cad**

IPv6 Configured IP Address: **fec0::d**

IPv6 Static IP Address:

IPv6 Prefix Length:

IPv6 Gateway Tunnel:

DNS Settings

Primary DNS:

Secondary DNS:

To change the IP address(es) of this device, make the applicable entries and click Apply. The "automatic" selection means DHCP. Changes to the IPv4 IP address will

take effect upon the next system restart.

If IPv6 is enabled, IPv6 will always have a Link-Local address, plus one configured address. The configured address will be either the static IP address, or an IPv6 address obtained from an IPv6 DHCP server. If no configured address appears, the DHCP server may have been unreachable.

The IPv6 static IP address window is the configured static address. If "Static" is selected and a new IP address entered as the static address, this new address will not take effect until the next system restart.

The numbers shown to the right of the IPv4 input windows are the actual numbers currently in use. If static IP addresses have been entered but the gateway has not been restarted yet, these numbers will not be the same.

You may use domain names instead of static IP addresses in several instances. If domain names are used, you must supply the IP address of at least one DNS server here. The DNS server must be at a static IP address. These changes take effect immediately. Note: If you are using DHCP, the DNS addresses will be supplied by the DHCP server and should be set to 0.0.0.0 here.

The Babel Buster maintains time and date via SNTP services.

Primary NTP Server	<input type="text" value="132.163.96.2"/>	Secondary NTP Server	<input type="text" value="129.6.15.30"/>
Daylight Time Start Rule	<input type="text" value="3.2.0/02:00:00"/>	Daylight Time End Rule	<input type="text" value="11.1.0/02:00:00"/>
Standard GMT Offset	<input type="text" value="-360"/> Minutes	Daylight GMT Offset	<input type="text" value="-300"/> Minutes
NTP Refresh Period	<input type="text" value="300"/> Minutes		
Current Local Time	<input type="text" value="2020-07-02 11:05:23"/> <input type="button" value="Refresh"/>		

NTP setup: Enter a primary and secondary IP address of NTP servers, such as those found at www.nist.gov (go to <http://tf.nist.gov/tf-cgi/servers.cgi> to find more). Enter daylight start/end rules, and offset from GMT for both standard and daylight time. Offset is a negative number in the western hemisphere. Enter an NTP update time in minutes. Do not set NTP to update too frequently or you risk being denied service by the NTP server. Click the Set NTP button after all settings have been made. The Flash update will take several seconds. The initial update of local time may take a minute or two. You may need to restart the Babel Buster if NTP had never before been initialized.

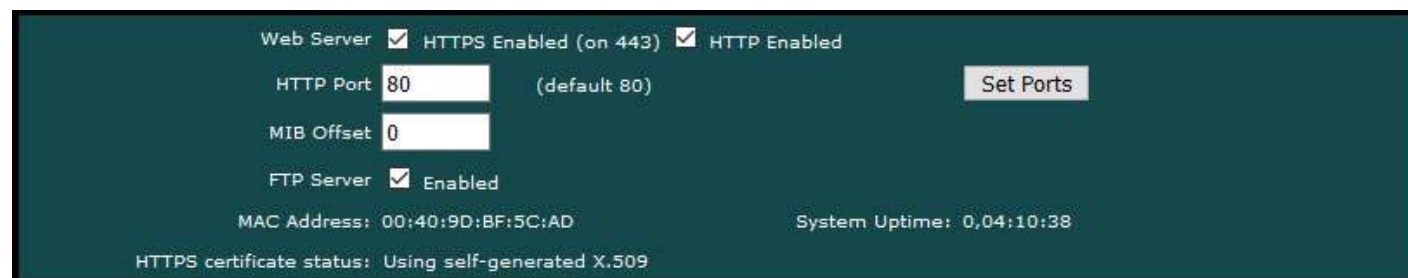
Daylight savings time start/end rules consist of "date/time" where the date (m.n.d) indicates the day when summer time starts or ends, and time (hour:min:sec) is the current local time when summer time starts/ends. The date portion of the rule is formatted as follows:

- m indicates the month ($1 \leq m \leq 12$)
- n indicates which week of the month ($1 \leq n \leq 5$). 5 = the last week in the month.
- d indicates what day of the week ($0 \leq d \leq 6$). 0 = Sunday

For example: Start "4.1.0/02:00:00", end "10.5.0/02:00:00" means summer time

starts at 2am on the first Sunday in April and ends at 2am on last Sunday in October. That was the old US rule. The new US rule is start "3.2.0/02:00:00" and end "11.1.0/02:00:00", which is start at 2am on the second Sunday in March, end at 2am on the first Sunday in November.

Note about time maintained here: SNMP and BACnet gateway functionality has no use for time and date. The only time you might have a need for valid time and date is when using SSL certificates for secure connections. If you are using a secure web connection and having trouble connecting, be sure NTP is set up here.



The screenshot shows a configuration window with a dark teal background. At the top, 'Web Server' is checked, followed by 'HTTPS Enabled (on 443)' and 'HTTP Enabled', both also checked. Below these are two input fields: 'HTTP Port' with the value '80' and '(default 80)' to its right, and 'MIB Offset' with the value '0'. A 'Set Ports' button is to the right of the HTTP Port field. Below the input fields, 'FTP Server' is checked and labeled 'Enabled'. At the bottom, the 'MAC Address' is '00:40:9D:BF:5C:AD' and 'System Uptime' is '0,04:10:38'. A status line at the very bottom reads 'HTTPS certificate status: Using self-generated X.509'.

Secure browsing can be enabled here, and non-secure can be disabled. You cannot disable both, and a forced configuration reset will restore HTTP (non-secure) web browsing. In order to use HTTPS, you must first upload the necessary SSL certificates (see Appendix G) or allow the certificates to be self-generated by explicitly deleting existing certificates.

IMPORTANT: It is highly recommended that in making the transition from HTTP to HTTPS, you enable both until you confirm HTTPS is functional. If there is a problem with the SSL certificates provided for HTTPS, then HTTPS will not run and you will find an error message on the "HTTPS certificate status" line. If you disable standard HTTP without first verifying that HTTPS is functional, you may end up locked out and will then need to do a forced hard reset (Appendix A.6).

The HTTP port for browsing the user interface can be moved away from the default HTTP port 80. Select a different port, click Set Ports, and then restart the gateway to make that new port take effect. Don't forget to append the port number to the gateway's IP address when attempting to browse the web user interface if it has been moved away from port 80.

The MIB offset lets you effectively move the entire MIB. This is required if more than one Babel Buster are going to be used on the same network but their configuration is different. Most SNMP managers do not know how to deal with MIBs that have the same set of OIDs but which mean different things in different devices using the "same" MIB. If the offset is changed, you will need to Reload SNMP to cause it to take effect.

FTP is enabled by default to allow firmware update uploads. It may be optionally disabled here. Just remember to enable it again before attempting a firmware update.

Any changes to this section (Set Ports button) require restarting the Babel Buster before they will take effect.

3.4 Resource Allocation

Historically, Control Solutions gateways had a fixed set of BACnet objects and other resources to work with. Invariably, there were always users that wanted less of this and more of that. Therefore, while there are still maximums imposed, you can now shift resources around as best suits your application. An example is shown below.

The values in the Pending column are those found in the most recently loaded XML configuration file. When saving or creating a new XML file, the numbers in the Current column will be written to the file. To change the allocations, change numbers in the Pending column. When you are ready to commit these changes, click the Commit button. To cause the changes to go into use, you must restart the device since memory allocation can occur only once at startup.

You can click the Check button prior to Commit to see if the values you have entered will be accepted. If adjustments need to be made, the values in the Pending column will be updated.

The first time you visit this page, you will see the initial default values. Should you change any of them, minimums and maximums currently defined in firmware will be imposed. If you see a value smaller than what you entered, it may be that you had exceeded the internal limit.

If you see that numbers toward the top of the list are large, and numbers near the bottom are all set to 1, it means the system has run out of free memory and you need to reallocate resources.

Local Objects	BACnet	SNMP	System
System Setup			
File Manager	Network	Resources	User

☐ Confirm

Resource	Current	Pending
Number of Analog Input Objects	100	<input type="text" value="100"/>
Number of Analog Output Objects	10	<input type="text" value="10"/>
Number of Analog Value Objects	10	<input type="text" value="10"/>
Number of Binary Input Objects	100	<input type="text" value="100"/>
Number of Binary Output Objects	10	<input type="text" value="10"/>
Number of Binary Value Objects	10	<input type="text" value="10"/>
Number of Multistate Input Objects	10	<input type="text" value="10"/>
Number of Multistate Output Objects	10	<input type="text" value="10"/>
Number of Multistate Value Objects	10	<input type="text" value="10"/>
Default States per Multistate Object	3	<input type="text" value="3"/>
Maximum COV Subscriptions	1000	<input type="text" value="1000"/>
Number of BACnet Client Devices	10	<input type="text" value="10"/>
Number of BACnet Client Read Maps	100	<input type="text" value="100"/>
Number of BACnet Client Write Maps	20	<input type="text" value="20"/>
MS/TP Slave Proxy Support	Disable	<input type="radio"/> Disable <input checked="" type="radio"/> Enable
Data Calculate Rule Count	100	<input type="text" value="100"/>
Data Copy Rule Count	100	<input type="text" value="100"/>
MIB Variable Count, Integer 32-bit	200	<input type="text" value="200"/>
MIB Variable Count, Float 32-bit	20	<input type="text" value="20"/>
Number of SNMP Trap Sender Devices	5	<input type="text" value="5"/>
Number of SNMP Trap Send Rules	50	<input type="text" value="50"/>
Number of SNMP Trap Receiver Devices	10	<input type="text" value="10"/>
Number of SNMP Trap Receive Rules	50	<input type="text" value="50"/>
Number of SNMP Client Devices	20	<input type="text" value="20"/>
Number of SNMP Client Read Rules	100	<input type="text" value="100"/>
Number of SNMP Client Write Rules	40	<input type="text" value="40"/>
Number of Table Walk Rules	10	<input type="text" value="10"/>
Estimated Memory Utilization	33.89%	33.89%

The estimated memory utilization shown at the bottom gives you an indication of how close you are to running out of memory. You will not be allowed to commit a resource allocation greater than 100%.

The object count limits for BACnet objects are set to 5,000 objects. However, if you were to try to allocate 5,000 of each type of object, you would never get there due to running out of resources.

3.5 User Login Passwords

There is only one default login provided initially, namely the username "root" with a unique password generated specifically for your particular Babel Buster. This password is provided to you in either external documentation included with the gateway, or it may be found on a label attached to the gateway. Network security laws in some jurisdictions require that Internet connected (or connectable) devices be shipped with unique default passwords, and the BB3-7302/MX-73 complies with this requirement.

Additional user logins may be created. The privilege level Administrator lets that user see and change anything. The privilege level Maintenance allows the user to log in and see (and change) values in the local objects via the Local Objects page, but cannot access any other pages. The Restricted level has no meaning in the BB3-7302/MX-73 (other than block access to everything) since it does not operate as a user defined web server.

You also have the option of IP filtering. If set, then the user can only access Babel Buster's web pages from that IP address. Leave set to 0.0.0.0 to disable filtering.

Local Objects		BACnet		SNMP		System	
System Setup							
File Manager		Network		Resources		User	
							Change
User Name	Password	Privilege Level	IP Filter	Confirm Change			
		Restricted ▾	0.0.0.0	<input type="checkbox"/>			
		Restricted ▾	0.0.0.0	<input type="checkbox"/>			
		Restricted ▾	0.0.0.0	<input type="checkbox"/>			
		Restricted ▾	0.0.0.0	<input type="checkbox"/>			
		Restricted ▾	0.0.0.0	<input type="checkbox"/>			
root	••••••	Unrestricted	0.0.0.0	<input type="checkbox"/>			
root confirm							



4. Configuring the BACnet Device

4.1 Device Object Parameters

The identity of the gateway as a BACnet device is entered on this page, along with other device object parameters.

The image shows the 'Babel Buster 3 BACNET ROUTER & GATEWAY' configuration interface. The top navigation bar includes 'Local Objects', 'BACnet', 'SNMP', and 'System'. Below this, a sub-navigation bar has 'Local Device', 'BACnet Client', 'Diagnostics', and 'BBMD'. The main content area is titled 'BACnet Device Settings:' and contains a 'Local Network Settings' section. This section includes fields for 'Device Instance' (208078), 'Port (default 0xBAC0 = 47808)' (47808), 'Device Object Name' (Babel Buster BB3-7302-V2C), 'Device Description', 'Device Location' (St. Paul, Minnesota), 'APDU Timeout' (3000), 'APDU Retries' (3), 'APDU Segment Timeout' (5000), 'Database Revision' (5), 'Local Command Priority' (10), 'Backlog Limit' (0), 'MS/TP Baud Rate' (38400), 'MS/TP MAC address' (78), 'MS/TP Max Info Frames' (10), and 'MS/TP Max Masters' (127). There are also checkboxes for 'Allow fault self-reset without Ack.' (checked), 'Disable Segmentation.' (unchecked), 'Disable self-restart upon communications loss' (checked), and a 'Slave Proxy Mode' section with radio buttons for 'None', 'Manual', and 'Auto'. A 'Save' button is located in the top right corner of the settings area.

Enter a device instance from 1 to 4,194,303. Enter a port number (note that 47808 is the standard port expected by most BIP devices).

The device object name, description, and location are entered here. The device object

name is expected to be unique to the entire BACnet network. Standard BACnet timeout and retry values are also entered on this page. These values are stored in a special area of non-volatile memory rather than the XML configuration file.

Local command priority is used when the result of a Calculate or Copy rule is written to a commandable object. It is also used if the result of a client read map is saved to a local commandable object, although this would not be recommended. Output objects are commandable. Client read maps should store results in input or value objects, while client write maps take their data from value or output objects. In other words, output objects should not be used for input.

Backlog Limit refers to the number of outstanding requests that the BACnet client will be allowed to make. The client has the ability to generate requests faster than they can be sent even at the 100 megabit rate, and may send them faster than the receiving device can process them. The client will generate up to this number of requests before pausing to wait for replies. The limit will default to the maximum of 10.

The MS/TP port parameters are set on this page. The MAC address should be in the range of 0 to 127. Max master setting must equal or exceed the highest used MAC address, and this setting must be the same in all devices on the MS/TP network.

Check the "allow" check box to allow faults to self-reset. These faults are those conditions indicated by a non-zero reliability code in any of the data objects (see list on data objects pages). Normally an external client needs to read the reliability code to acknowledge the fault before it will automatically reset. By checking the "allow" check box, faults will automatically self-reset without acknowledgement. This is required any time the client does not periodically read reliability codes but does check fault status - a behavior known to be common to BMS front ends.

Check the "disable self-restart" box to disable self restart upon communication loss. If this box is not checked, this gateway will restart itself in an attempt to auto-recover if communications with devices has started and then stopped.

Segmentation can be disabled. This would only be valid when the Babel Buster is used as a gateway and the client does not support segmentation. As a router, this setting has no bearing on whether other devices may try to use segmentation.

If you have MS/TP devices on the local MS/TP link that are slave-only, i.e. do not respond to Who-Is, then you will want to enable Slave Proxy Mode. Use the Manual setting if you will enter those devices explicitly on the Slave Proxy page. Select Auto mode to attempt to auto-discover them.

Click Save to store. This store process will take a little while as these parameters are being saved to non-volatile memory. A change in port number will not take effect until the next system restart.

4.2 Network Settings

The two most important things that must be unique on the BACnet IP network are

device instance, and IP address. The IP address is set on the Network page.

Babel Buster[®] 3
BACNET
ROUTER & GATEWAY
MODEL BB3-7302-V2C

CONTROL SOLUTIONS MINNESOTA

Local Objects BACnet Modbus System System Setup

File Manager **Network** Resources User

IPv4 Settings ☐ Automatic ☒ Static

IPv4 Static IP Address IPv4 Configured IP Address **192.168.1.126**

IPv4 Static Subnet Mask IPv4 Subnet Mask **255.255.255.0**

IPv4 Static Gateway IPv4 Gateway **192.168.1.1**

IPv6 Settings ☐ Disabled ☒ Automatic ☐ Static

IPv6 Link-Local IP Address **fe80::240:9dff:fe45:4696**

Select either Static or Automatic for IPv4. To change the Static IP address of this device, enter the address, subnet mask, and gateway, then click Apply. Select Automatic to specify that DHCP should be used to obtain an IP address upon power-up. IP address change will take effect upon next power cycle.

The above screen shot is only a portion of the Network setup page, and is the only part of the Network page that is required for BACnet IP. The remainder of the Network page is discussed in Section 3.3.

The Web User Interface is accessible via IPv6; however, the BACnet IP Client does not yet support IPv6. The only demonstrated version of BACnet IP over IPv6 does not use actual IPv6 addresses - it uses Virtual MAC addresses (VMAC) and address translation tables. The VMAC approach allows IPv6 to coexist with original IPv4 devices. As of the BTL testing of this device, test specifications for BACnet IP over IPv6 were not available and thus IPv6 support is not included in this device.



5. Configuring the BACnet Router

5.1 Router Settings

The local network numbers are set here.

The screenshot shows the 'Router Settings' tab in the Babel Buster 3 software. The interface has a dark green header with the product name and logo. Below the header is a navigation bar with tabs: 'Local Objects', 'BACnet', 'SNMP', 'System', 'Local Device', 'BACnet Client', 'Diagnostics', 'BBMD', 'BACnet Settings', 'Router Settings' (selected), 'Configured Routes', 'Discovered Routes', and 'Slave Proxy'. The 'Local Router Settings' section contains four input fields: 'BACnet IP Network Number' (set to 8), 'Hop Count' (set to 10), 'BACnet IP Network Info' (set to 'Default IP Net'), 'MS/TP Network Number' (set to 115), 'Hop Count' (set to 10), and 'MS/TP Network Info' (set to 'Default MS/TP Net'). There is a 'Save' button to the right of the first two fields. At the bottom, there are three checkboxes: 'Enable Router' (checked), 'Enable I-Am route learning' (checked), and 'Enable I-Am-Router route learning' (checked).

Enter the network numbers for the networks directly connected to this router. These must be unique across the entire internetwork. These numbers will ultimately tell other routers how to find devices on these local networks.

The Info strings are strictly informational and have no bearing on functionality.

Hop count should be the maximum number of routers on the network. Hop count limits the number of times a packet is forwarded by a router. It should not be forwarded more times than there are routers, but if a router is misconfigured, a message may bounce back and forth until the hop count expires (hop count defaults to 255). By setting the hop count equal to the number of routers downstream, a message being bounced by misconfigured routers will expire sooner than later and not

jam the network.

Router functionality can be disabled when the Babel Buster should operate only as a gateway. Multiple routers across the same networks will cause network failure. Disable router functionality any time routing will not be used.

Networks can be configured or learned. To disable learning in conjunction with configuring network routes, make the applicable selections above.

5.2 Configured Routes

Networks and other routers can be automatically learned or they can be configured. If route learning is disabled, then you must provide routing information to other networks here.

Local Device		BACnet Client		Diagnostics		BBMD			
BACnet Settings		Router Settings		Configured Routes		Discovered Routes		Slave Proxy	
Showing 1 to 15 of 64									
<input type="button" value="Update"/> <input type="button" value=" < Prev"/> <input type="button" value=" Next >"/>									
Local Port	Remote Network #	Local Info				Router's Address			
BACnet IP ▾	41	Network 41				192.168.1.178:47808			
MS/TP ▾	88	Network 88				91			
None ▾	0					---			

Enter the known remote network numbers and the ports via which they may be reached.

The Info strings are strictly informational and have no bearing on functionality.

The router's address is optional. If not given here, it will be searched for on the network using Who-Is-Router. You may enter the remote router's address as an MS/TP MAC address, or an IP address optionally with port number. If no port number is given, the BB3-7302/MX-73's own local port number will be used. IP should be given in the form of 192.168.1.199:47808 (for example) or just 192.168.1.199.

It should be noted that even if you do enter the router's address here, it will be replaced in the event an I-Am-Router message is received for the given network number but having a different router address.

It should also be noted that if some external BACnet network management tool sends a router table initialize message to this device, the entire page shown here will be replaced. After a delay of a few minutes, the new contents of this page will be auto-saved to the XML configuration file for subsequent reload. Thus, the router portion of this BB3-7302/MX-73 may be remotely managed.

If you leave this page completely empty, all routers needed for routing of traffic will be located using the Who-Is-Router broadcast to the network. Routers that are found this way, in addition to any listed here, will be listed on the Discovered page.

5.3 Discovered Routes

The summary of both learned and discovered routes is listed on this page.

Local Device					
BACnet Client					
Diagnostics					
BBMD					
BACnet Settings					
Router Settings					
Configured Routes					
Discovered Routes					
Slave Proxy					
Showing 1 to 15 of 64					
Update < Prev Next >					
Local Port	Remote Network #	Local Info		Router's Address	
BACnet IP	41	Network 41		192.168.1.178:47808	
MS/TP	88	Network 88		91	
BACnet IP	78			192.168.1.78:47808	
BACnet IP	160			192.168.1.168:47808	
BACnet IP	79			192.168.1.79:47808	
BACnet IP	19			192.168.1.178:47808	

Use the Prev/Next buttons to scroll through the list of known networks. This list is a combination of configured networks and those discovered via the Who-Is-Router and I-Am-Router message exchange.

The Local Info column will be populated using the information provided on the Configured Routes page. If the route is automatically learned, local information is not provided in the response to Who-Is-Router, and therefore this column remains blank. The local information does not affect operation in any way.

5.4 Slave Proxy

This page allows you to configure slave proxy support to enable auto-discovery of MS/TP slave devices which do not respond to Who-Is requests. This router will respond with an I-Am message on behalf of the device(s) listed here when a Who-Is is received.

Local Device					
BACnet Client					
Diagnostics					
BBMD					
BACnet Settings					
Router Settings					
Configured Routes					
Discovered Routes					
Slave Proxy					
Showing 1 to 2 of 2					
Update < Prev Next >					
Mode	MAC Address	Device Instance	Max APDU	Vendor ID	Local Device Info
M	181	20821	480	208	BB2-3010 Slave 21
M	201	20841	480	208	BB2-3010 Slave 41
	0	0			
Add Remove Convert					

Enable Slave Proxy Mode on the BACnet Settings page. If Manual mode is selected, then enter the slave device information for a given slave, and click Add. Repeat the Add for each slave to be manually configured. If Auto mode is selected, then you

can disregard the Add. If you wish to remove a previously configured (or auto configured) slave, enter just the MAC address and then click Remove.

If you have initially selected Auto mode and wish to capture the discovered slaves and convert them to Manual mode entries, click the Convert button. After clicking Convert and saving the configuration file on the File Manager page, you may then switch to Manual mode to reduce overhead on the MS/TP link. Scanning through the entire list of possible slave addresses (polling each to see if present) can take as long as a couple of minutes.

The local device info is not used on the network. It is simply present to aid in documenting the proxies.

Use the Update, Prev, and Next buttons to view the list of currently configured slaves for which proxy support is configured.

The Mode in the first column will indicate the following:

M = Added using Add button above, or by writing to Device Object

Manual_Slave_Address_Binding property

U = Added as for 'M' but not responding

X = Added as for 'M' but found to execute Who-Is (is not a slave)

A = Discovered by Automatic slave proxy mode

S = Added by writing to Device Object Slave_Address_Binding property (temporary binding)



6. Configuring Local Objects

Babel Buster gateways do not come with a predefined set of BACnet objects. The gateway will initially have a handful of objects, but it is up to the user to allocate the number needed, up to the maximum permitted by available resources.

6.1 Behavior of Input vs Output Objects

The easiest way to keep track of input versus output is to think about a BACnet device's role in the system. The system will receive input from the BACnet device, and provide output to the BACnet device. Inside the BACnet device, hardware will physically associate BACnet Input Objects with sensor inputs such as temperature or pressure sensors, etc. The system then receives the sensor input information via BACnet Input Objects. When the system wants to control an actuator, it will send setpoints to the actuator via BACnet Output Objects. Hardware inside the BACnet device will physically associate the Output Object with a physical actuator such as valve position servo or motor speed controller.

Keeping track of input versus output in a gateway can be a bit trickier; however, the choice of input versus output does not change from the BACnet perspective. Only the nature of the physical sensor and actuator hardware changes. In the case of the BB3-7302/MX-73, sensors and actuators both consist of SNMP devices (from BACnet's perspective) or other BACnet devices. Therefore, use a BACnet Output Object to send data to another device acting as an actuator, and use a BACnet Input Object to receive data from another device acting as a sensor.

We have not mentioned BACnet Value Objects yet just to avoid confusing the discussion. A Value Object can be input or output, or both at the same time. When using a Value Object, it is best to think about its role as input or output when deciding how to apply maps or rules in the gateway.

6.2 Allocating Local Objects

The resource allocation page is where you set the number of each type of available BACnet object that you will use. It is a good idea to determine ahead of time how many objects you will need, then allocate that number, possibly including a spare object or two. It is not a good idea to allocate a large number of objects that will remain unused since this simply clutters the screen when a front end system auto-discovers all objects in the device.

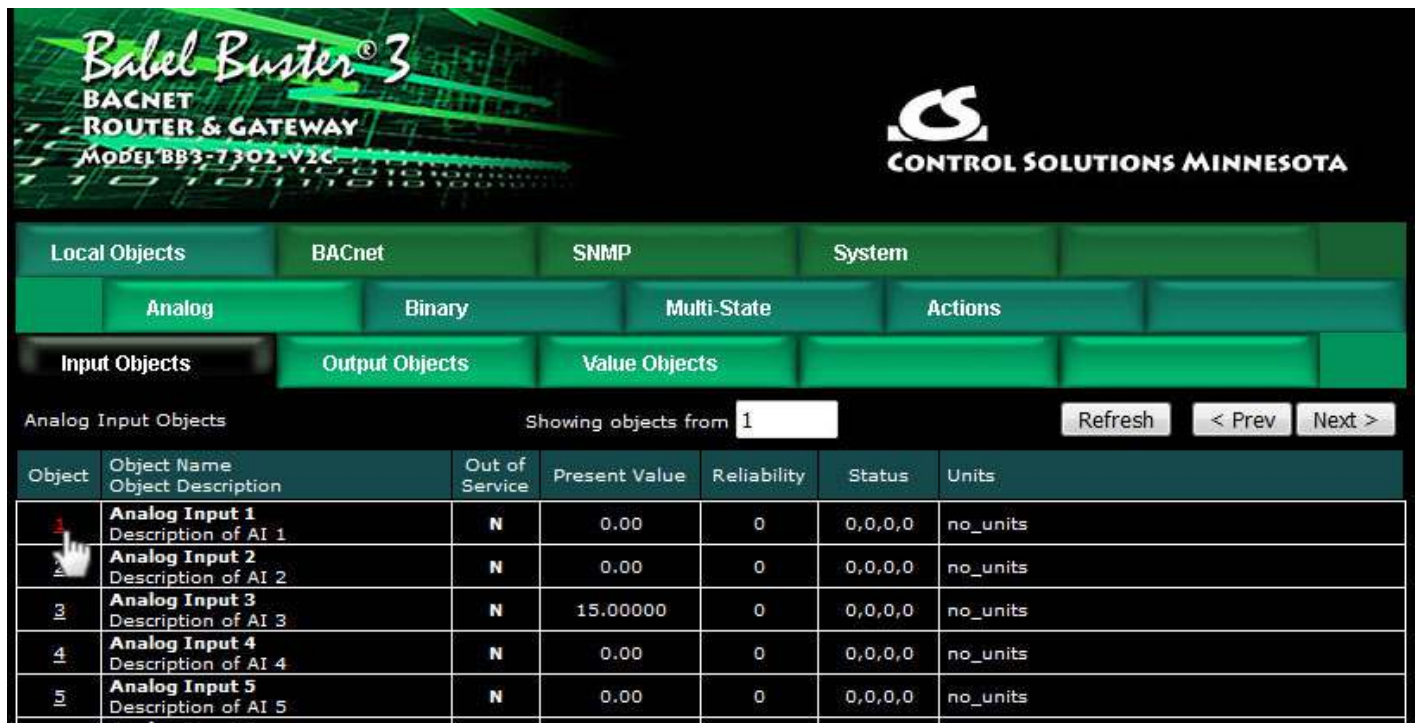
Local Objects	BACnet	SNMP	System	
System Setup				
File Manager	Network	Resources	User	
<div>Check Commit <input type="checkbox"/> Confirm Restart</div>				
Resource	Current	Pending		
Number of Analog Input Objects	100	<input type="text" value="100"/>		
Number of Analog Output Objects	10	<input type="text" value="10"/>		
Number of Analog Value Objects	10	<input type="text" value="10"/>		
Number of Binary Input Objects	100	<input type="text" value="100"/>		
Number of Binary Output Objects	10	<input type="text" value="10"/>		
Number of Binary Value Objects	10	<input type="text" value="10"/>		
Number of Multistate Input Objects	10	<input type="text" value="10"/>		
Number of Multistate Output Objects	10	<input type="text" value="10"/>		
Number of Multistate Value Objects	10	<input type="text" value="10"/>		
Default States per Multistate Object	3	<input type="text" value="3"/>		

The portion of the Resources page dedicated to BACnet object counts is shown above. For a complete discussion of the Resources page, including how to change the counts, refer to Section 3.4 in this User Guide.

6.3 Configuring Local Objects

There is a different web page for each BACnet object type in the device. Objects are listed in tabular form with name and description, present value, reliability code and status. Additional information as applicable to the object type may also be listed.

Click on the object number in the first column to open the expanded view of that object and gain access to its configuration.



Babel Buster[®] 3
BACNET
ROUTER & GATEWAY
MODEL BB3-7302-V2C

CONTROL SOLUTIONS MINNESOTA

Local Objects | BACnet | SNMP | System

Analog | Binary | Multi-State | Actions

Input Objects | Output Objects | Value Objects

Analog Input Objects Showing objects from 1 Refresh < Prev Next >

Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Units
1	Analog Input 1 Description of AI 1	N	0.00	0	0,0,0,0	no_units
2	Analog Input 2 Description of AI 2	N	0.00	0	0,0,0,0	no_units
3	Analog Input 3 Description of AI 3	N	15.000000	0	0,0,0,0	no_units
4	Analog Input 4 Description of AI 4	N	0.00	0	0,0,0,0	no_units
5	Analog Input 5 Description of AI 5	N	0.00	0	0,0,0,0	no_units

Reliability codes indicate error conditions if non-zero, and may be any of the following:

- BACnet client, device timeout (82)
- BACnet client, error returned by server (83)
- SNMP client, no response from agent (91)
- SNMP client, agent returned 'no such name' error (92)
- SNMP client, unable to parse data returned (93)
- SNMP client, reply did not match request (94)
- SNMP client, invalid table walk configuration (95)
- SNMP client, unable to parse PDU returned (96)
- SNMP client, table walk came up short on data (97)
- SNMP client, agent returned error other than 'no such name' (98)

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

- A = in alarm
- B = fault
- C = overridden
- D = out of service

6.3.1 Analog Input Objects

The source of data for an Analog Input object will typically be reading from some other BACnet or SNMP device.

The screenshot shows a web-based configuration interface for 'Local Objects'. The top navigation bar includes tabs for 'Local Objects', 'BACnet', 'SNMP', and 'System'. Below this, there are sub-tabs for 'Analog', 'Binary', 'Multi-State', and 'Actions'. The 'Input Objects' tab is currently selected, showing 'Analog Input # 1'. To the right of the tab are 'Update', '< Prev', and 'Next >' buttons. The main configuration area for 'Analog Input # 1' includes the following fields and controls:

- Reliability: 0
- Status: 0,0,0,0
- Device Link: RTU_R1
- Out of Service: ☐
- Deconfigure: ☐
- Object name: Analog Input 1
- Force: ☐
- Present Value: 0.00
- Description: Description of AI 1
- COV increment: 0.00
- Units: no_units (selected from a dropdown menu)

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page, or Update to accept changes.

The object name and description may be changed here. BACnet units may be selected. Initial COV increment may be entered. When any of these are changed, be sure to save the updated configuration by executing "Save XML Config File" on the File Manager page.

The object may be set Out of Service by checking that box and clicking Update. The present value may be changed by entering a value, checking Force, and clicking Update.

The source of data for an Analog Input object will typically be reading from some other BACnet or SNMP device via the map indicated by the Device Link. The mapped device will be polled at the rate specified by the Read Map.

Out of Service means polling of the mapped remote device will stop. While out of service, the present value may be written by the BACnet client. Data may be forced via this web page at any time, but will be overwritten by the next mapped client update unless the object is out of service.

Reliability codes indicate errors as itemized on the tabular object list.

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

- A = in alarm
- B = fault
- C = overridden
- D = out of service

Device link will indicate BIP or SNMP, followed by R for read or W for write, and a number which is the map number in the table of read or write maps for mapping to this BACnet object. The designation R means read from a remote device, and W means write to a remote device.

Check 'Deconfigure' and click Update to erase configuration for this object.

6.3.2 Analog Output Objects

The destination of data for an Analog Output object will typically be some other BACnet or SNMP device.

The screenshot shows a web-based configuration interface for BACnet objects. At the top, there are tabs for 'Local Objects', 'BACnet', 'SNMP', and 'System'. Below these are sub-tabs for 'Analog', 'Binary', 'Multi-State', and 'Actions'. A third row of tabs includes 'Input Objects', 'Output Objects' (which is selected), and 'Value Objects'. The main content area is titled 'Analog Output # 1'. It contains several fields: 'Reliability: 0', 'Status: 0,0,0,0', 'Device Link: ---', 'Out of Service: ☐', and 'Deconfigure: ☐'. There is a text field for 'Object name' containing 'Analog Output 1', a 'Force' checkbox, a 'Present Value' field with '0.00', and a priority level dropdown set to 'rq> 0.00'. Below these are 'Description' (containing 'Description of AO 1'), 'COV increment' (0.00), 'Relinquish Default' (0.00), and a 'Units' dropdown set to 'no_units'. At the bottom right are buttons for 'Update', '< Prev', and 'Next >'.

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page, or Update to accept changes.

The object name and description may be changed here. BACnet units may be selected. Initial COV increment may be entered. When any of these are changed, be sure to save the updated configuration by executing "Save XML Config File" on the File Manager page.

The destination of data for an Analog Output object will be writing the remote BACnet or SNMP device via the map indicated by the Device Link. The remote device will be updated upon change of source data and/or periodically as defined by the Write Map.

The Analog Output object is commandable, meaning the BACnet client must write both a value and a priority level for that value. The highest level value will be the one written to the remote device (if one is mapped). If all values are relinquished, the relinquish default value will be written to the remote device.

To set an output object manually from this page, check the Force box, enter a value in the Present Value window, and select a priority level to assign to your forced value. Then click Update. To return a given priority level to NULL, simply type the word NULL in the Present Value window, check Force, and click Update.

Out of service means the mapped remote device will not be written to. Values written by the BACnet client will be retained, but only applied when this object is placed back in service. At that time, the highest priority value will be written to the remote device.

Reliability codes indicate errors as itemized on the tabular object list.

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

A = in alarm

B = fault

C = overridden

D = out of service

Device link will indicate BIP or SNMP, followed by R for read or W for write, and a number which is the map number in the table of read or write maps for mapping to this BACnet object. The designation R means read from a remote device, and W means write to a remote device.

Check 'Deconfigure' and click Update to erase configuration for this object.

6.3.3 Analog Value Objects

Analog Value objects may be both a source and destination for some other BACnet or SNMP device.

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page, or Update to accept changes.

The object name and description may be changed here. BACnet units may be selected. Initial COV increment may be entered. When any of these are changed, be sure to save the updated configuration by executing "Save XML Config File" on the File Manager page.

The object may be set Out of Service by checking that box and clicking Update. The present value may be changed by entering a value, checking Force, and clicking Update.

Analog Value objects may be both a source and destination for some other BACnet or SNMP device. The source of data for the Value object will be reading from a remote device when associated with a Read Map. The destination of data for the Value object will be writing to a remote device when associated with a Write Map. If a remote device is mapped, the device links are displayed above. You may click on either link to view the respective mapping.

The Value object may be simultaneously associated with both Read and Write maps pointing to the same remote device object. When this Value object receives new data (from any source), this data will be written to the mapped remote device before any subsequent read from the same device. Thus the Value data is not discarded by the read operation before the new data can be written.

Out of Service means polling of the remote device will stop. While out of service, the present value may be written by an external BACnet client but it will not be written to any mapped remote device. Data may be forced via this web page at any time, but will be overwritten by the next read from a remote device unless the object is out of service.

Reliability codes indicate errors as itemized on the tabular object list.

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

A = in alarm

B = fault

C = overridden

D = out of service

Device link will indicate BIP or SNMP, followed by R for read or W for write, and a number which is the map number in the table of read or write maps for mapping to this BACnet object. The designation R means read from a remote device, and W means write to a remote device.

Check 'Deconfigure' and click Update to erase configuration for this object.

6.3.4 Binary Input Objects

The source of data for a Binary Input object will typically be reading from some other BACnet or SNMP device.

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page, or Update to accept changes.

The object name and description may be changed here. State text may be entered. When any of these are changed, be sure to save the updated configuration by executing "Save XML Config File" on the File Manager page.

The object may be set Out of Service by checking that box and clicking Update. The present value may be changed by entering a value, checking Force, and clicking Update.

The source of data for an Binary Input object will typically be reading from some other BACnet or SNMP device via the map indicated by the Device Link. The mapped device will be polled at the rate specified by the Read Map.

Out of Service means polling of the mapped remote device will stop. While out of service, the present value may be written by the BACnet client. Data may be forced via this web page at any time, but will be overwritten by the next mapped client update unless the object is out of service.

Reliability codes indicate errors as itemized on the tabular object list.

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

A = in alarm

B = fault

C = overridden

D = out of service

Device link will indicate BIP or SNMP, followed by R for read or W for write, and a number which is the map number in the table of read or write maps for mapping to this BACnet object. The designation R means read from a remote device, and W means write to a remote device.

Check 'Deconfigure' and click Update to erase configuration for this object.

6.3.5 Binary Output Objects

The destination of data for a Binary Output object will typically be some other BACnet or SNMP device.

The screenshot shows a web interface for configuring Binary Output objects. At the top, there are tabs for 'Local Objects', 'BACnet', 'SNMP', and 'System'. Under 'Local Objects', there are sub-tabs for 'Analog', 'Binary', 'Multi-State', and 'Actions'. The 'Binary' tab is selected, and under it, 'Output Objects' is highlighted. The main content area shows the configuration for 'Binary Output # 1'. It includes fields for 'Reliability: 0', 'Status: 0,0,0,0', 'Device Link: ---', 'Out of Service: ☐', and 'Deconfigure: ☐'. There are also buttons for 'Update', '< Prev', and 'Next >'. The 'Object name' field contains 'Binary Output 1', and the 'Description' field contains 'Description of BO 1'. The 'Active Text' field contains 'Binary output is Active', and the 'Inactive Text' field contains 'Binary output is Inactive'. The 'Relinquish Default' dropdown is set to 'Inactive'. The 'Force' checkbox is unchecked, and the 'Present Value' dropdown is set to 'Inactive'.

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page, or Update to accept changes.

The object name and description may be changed here. State text may be entered. When any of these are changed, be sure to save the updated configuration by executing "Save XML Config File" on the File Manager page..

The destination of data for a Binary Output object will be writing the remote BACnet or SNMP device via the map indicated by the Device Link. The remote device will be updated upon change of source data and/or periodically as defined by the Write Map.

The Binary Output object is commandable, meaning the BACnet client must write both a value and a priority level for that value. The highest level value will be the one written to the remote device (if one is mapped). If all values are relinquished, the relinquish default value will be written to the remote device.

To set an output object manually from this page, check the Force box, enter a value in the Present Value window, and select a priority level to assign to your forced value. Then click Update. To return a given priority level to NULL, simply type the word NULL in the Present Value window, check Force, and click Update.

Out of service means the mapped remote device will not be written to. Values written by the BACnet client will be retained, but only applied when this object is placed back in service. At that time, the highest priority value will be written to the remote device.

Reliability codes indicate errors as itemized on the tabular object list.

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

A = in alarm

B = fault

C = overridden

D = out of service

Device link will indicate BIP or SNMP, followed by R for read or W for write, and a number which is the map number in the table of read or write maps for mapping to this BACnet object. The designation R means read from a remote device, and W means write to a remote device.

Check 'Deconfigure' and click Update to erase configuration for this object.

6.3.6 Binary Value Objects

Binary Value objects may be both a source and destination for some other BACnet or SNMP device.

Local Objects | BACnet | SNMP | System

Analog | Binary | Multi-State | Actions

Input Objects | Output Objects | Value Objects

Binary Value # 1 [Update] [< Prev] [Next >]

Reliability: 0 Status: 0,0,0,0 Device Link: --- --- Out of Service: ☐ Deconfigure: ☐

Object name: Binary Value 1 Force: ☐ Present Value: Inactive

Description: Description of BV 1

Active Text: Binary value is Active Inactive Text: Binary value is Inactive

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page, or Update to accept changes.

The object name and description may be changed here. State text may be entered. When any of these are changed, be sure to save the updated configuration by executing "Save XML Config File" on the File Manager page.

The object may be set Out of Service by checking that box and clicking Update. The present value may be changed by entering a value, checking Force, and clicking Update.

Binary Value objects may be both a source and destination for some other BACnet or SNMP device. The source of data for the Value object will be reading from a remote device when associated with a Read Map. The destination of data for the Value object will be writing to a remote device when associated with a Write Map. If a remote device is mapped, the device links are displayed above. You may click on either link to view the respective mapping.

The Value object may be simultaneously associated with both Read and Write maps pointing to the same remote device object. When this Value object receives new data (from any source), this data will be written to the mapped remote device before any subsequent read from the same device. Thus the Value data is not discarded by the read operation before the new data can be written.

Out of Service means polling of the remote device will stop. While out of service, the present value may be written by an external BACnet client but it will not be written to any mapped remote device. Data may be forced via this web page at any time, but will be overwritten by the next read from a remote device unless the object is out of service.

Reliability codes indicate errors as itemized on the tabular object list.

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

A = in alarm

B = fault

C = overridden

D = out of service

Device link will indicate BIP or SNMP, followed by R for read or W for write, and a number which is the map number in the table of read or write maps for mapping to this BACnet object. The designation R means read from a remote device, and W means write to a remote device.

Check 'Deconfigure' and click Update to erase configuration for this object.

6.3.7 Multistate Input Objects

The source of data for a Multistate Input object will typically be reading from some other BACnet or SNMP device.

The screenshot shows a web-based configuration interface for 'Local Objects'. The top navigation bar includes tabs for 'Local Objects', 'BACnet', 'SNMP', and 'System'. Under 'Local Objects', there are sub-tabs for 'Analog', 'Binary', 'Multi-State', and 'Actions'. The 'Multi-State' sub-tab is active, showing 'Input Objects', 'Output Objects', and 'Value Objects'. The 'Input Objects' sub-tab is selected, displaying a list of objects. The first object, 'Multi-State Input # 1', is highlighted. Below the list, the configuration details for this object are shown: Reliability: 0, Status: 0,0,0,0, Device Link: ---, Out of Service: ☐, Deconfigure: ☐, Object name: Multi-state Input 1, Force: ☐, Present Value: 1, Description: Description of MI 1, Maximum State Value: 3, Value: 1, Text: (empty), and an 'Add/Change' button. Below these fields, the state text for this object is listed: 1: Low, 2: Medium, 3: High. Navigation buttons 'Update', '< Prev', and 'Next >' are located at the top right of the configuration area.

The object name, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page, or Update to accept changes.

The object name and description may be changed here. When changed, be sure to save the updated configuration by executing "Save XML Config File" on the File Manager page.

State text may be added. Before adding text, set the maximum state value for this object. Then add text strings corresponding to each of the number of states allocated by entering the value, corresponding text, and clicking Add/Change. When changed, be sure to save the updated configuration by executing "Save XML Config File" on the File Manager page.

The object may be set Out of Service by checking that box and clicking Update. The present value may be changed by entering a value, checking Force, and clicking Update.

The source of data for a Multistate Input object will typically be reading from some other BACnet or SNMP device via the map indicated by the Device Link. The mapped device will be polled at the rate specified by the Read Map.

Out of Service means polling of the mapped remote device will stop. While out of service, the present value may be written by the BACnet client. Data may be forced via this web page at any time, but will be overwritten by the next mapped client update unless the object is out of service.

Reliability codes indicate errors as itemized on the tabular object list.

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

- A = in alarm
- B = fault
- C = overridden
- D = out of service

Device link will indicate BIP, or SNMP, followed by R for read or W for write, and a number which is the map number in the table of read or write maps for mapping to this BACnet object. The designation R means read from a remote device, and W means write to a remote device.

Check 'Deconfigure' and click Update to erase configuration for this object.

6.3.8 Multistate Output Objects

The destination of data for a Multistate Output object will typically be some other BACnet or SNMP device.

The object name, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page, or Update to accept changes.

The object name and description may be changed here. When changed, be sure to save the updated configuration by executing "Save XML Config File" on the File Manager page.

State text may be added. Before adding text, set the maximum state value for this object. Then add text strings corresponding to each of the number of states allocated by entering the value, corresponding text, and clicking Add/Change. When changed, be sure to save the updated configuration by executing "Save XML Config File" on the File Manager page.

The object may be set Out of Service by checking that box and clicking Update. The present value may be changed by entering a value, checking Force, and clicking Update.

The Multistate Output object is commandable, meaning the BACnet client must write both a value and a priority level for that value. The highest level value will be the one written to the remote device (if one is mapped). If all values are relinquished, the relinquish default value will be written to the remote device.

To set an output object manually from this page, check the Force box, enter a value in the Present Value window, and select a priority level to assign to your forced value. Then click Update. To return a given priority level to NULL, simply type the word NULL in the Present Value window, check Force, and click Update.

Out of service means the mapped remote device will not be written to. Values written by the BACnet client will be retained, but only applied when this object is placed back in service. At that time, the highest priority value will be written to the remote device.

Reliability codes indicate errors as itemized on the tabular object list.

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

A = in alarm

B = fault

C = overridden

D = out of service

Device link will indicate BIP or SNMP, followed by R for read or W for write, and a number which is the map number in the table of read or write maps for mapping to this BACnet object. The designation R means read from a remote device, and W means write to a remote device.

Check 'Deconfigure' and click Update to erase configuration for this object.

6.3.9 Multistate Value Objects

Multistate Value objects may be both a source and destination for some other BACnet or SNMP device.

The screenshot shows a web-based configuration interface for Multistate Value Objects. At the top, there are tabs for 'Local Objects', 'BACnet', 'SNMP', and 'System'. Below these are sub-tabs for 'Analog', 'Binary', 'Multi-State', and 'Actions'. The 'Multi-State' tab is selected, and within it, the 'Value Objects' sub-tab is active. The main content area displays the configuration for 'Multi-State Value # 1'. It includes fields for 'Reliability: 0', 'Status: 0,0,0,0', 'Device Link: --- ---', 'Out of Service' (checkbox), and 'Deconfigure' (checkbox). The 'Object name' is 'Multi-state Value 1', and the 'Description' is 'Description of MV 1'. The 'Force' checkbox is checked, and the 'Present Value' is set to '1'. The 'Maximum State Value' is '3'. There are 'Update', '< Prev', and 'Next >' buttons. At the bottom, there is a section for 'State text for this object' with three entries: '1: ---', '2: ---', and '3: ---'. An 'Add/Change' button is also present.

The object name, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page, or Update to accept changes.

The object name and description may be changed here. When changed, be sure to save the updated configuration by executing "Save XML Config File" on the File

Manager page.

State text may be added. Before adding text, set the maximum state value for this object. Then add text strings corresponding to each of the number of states allocated by entering the value, corresponding text, and clicking Add/Change. When changed, be sure to save the updated configuration by executing "Save XML Config File" on the File Manager page.

The object may be set Out of Service by checking that box and clicking Update. The present value may be changed by entering a value, checking Force, and clicking Update.

Multistate Value objects may be both a source and destination for some other BACnet or SNMP device. The source of data for the Value object will be reading from a remote device when associated with a Read Map. The destination of data for the Value object will be writing to a remote device when associated with a Write Map. If a remote device is mapped, the device links are displayed above. You may click on either link to view the respective mapping.

The Value object may be simultaneously associated with both Read and Write maps pointing to the same remote device object. When this Value object receives new data (from any source), this data will be written to the mapped remote device before any subsequent read from the same device. Thus the Value data is not discarded by the read operation before the new data can be written.

Out of Service means polling of the remote device will stop. While out of service, the present value may be written by an external BACnet client but it will not be written to any mapped remote device. Data may be forced via this web page at any time, but will be overwritten by the next read from a remote device unless the object is out of service.

Reliability codes indicate errors as itemized on the tabular object list.

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

A = in alarm

B = fault

C = overridden

D = out of service

Device link will indicate BIP or SNMP, followed by R for read or W for write, and a number which is the map number in the table of read or write maps for mapping to this BACnet object. The designation R means read from a remote device, and W means write to a remote device.

Check 'Deconfigure' and click Update to erase configuration for this object.

6.4 Local Object Calculate Rules

The Babel Buster BB3-7302/MX-73 includes the ability to do simple calculations based on simple template rules. Select the operation, one or two operands as applicable, and a object to place the result in. Operations like "multiply" will use objects A "and" B.

Operations like "sum" can add up the contents of a series of objects by selecting "thru" instead of "and". These template rules can be useful for doing minor data manipulation or testing for purposes of enabling rules, or for generating derived values.

Rule #	Perform Operation	Using Object	And/Thru Using	This Object/Value	Place Result in Object
1	none	None	and	None	None

Rules Enabled: 1

Buttons: Insert, Delete

Here is an example of a template rule that multiplies the value of Analog Input 7 by value of Analog Input 8 and places the result in Analog Input 15. An example of application would be to multiply a voltage reading input by a current reading input to derive a power value presented as an input.

Rule #	Perform Operation	Using Object	And/Thru Using	This Object/Value	Place Result in Object
1	multiply	AI 7	and	AI 8	AI 15
2	none	None	and	None	None

Showing 1 to 2 of 2

Rules Enabled: 2

Buttons: Insert, Delete

There are times when you may want to make a calculation based on a constant value. One way of introducing a constant into a calculation is by reserving a commandable object to hold that constant, and then configuring the relinquish default to be that value. Then reference that object in your calculate rule.

The "set" operation can be used to directly place an unsigned integer value into an object. The screen shot below illustrates setting Analog Value 1 to a value of 12345. The value in a set operation can only be unsigned integer as the value was originally

intended for use in bit mask operations.

Calculate		Copy		Report			
Showing 1 to 2 of 2						Update	< Prev Next >
Rule #	Perform Operation	Using Object	And/Thru Using	This Object/Value	Place Result in Object		
1	set	AV 1	using	12345	AV 1		
2	none	None	and	None	None		
# Rules Enabled: 2						Insert	Delete

Operations available on two or more objects using 'and' or 'thru':

add	Add two objects
average	Average two or more objects
sum	Sum two or more objects
subtract	Subtract second object from first
multiply	Multiply two objects
divide	Divide first object by second
logic OR	Logically OR two or more objects
logic AND	Logically AND two or more objects
logic NOR	Logically NOR two or more objects
logic NAND	Logically NAND two or more objects
logic XOR	Logically Exclusive-OR two objects

Operations available on one object:

logic NOT	Generate bit-wise negation of object
test = 0	Set result to 'true' if object is zero
test < 0	Set result to 'true' if object is less than zero
test > 0	Set result to 'true' if object is greater than zero
relinquish	Relinquish command priority previously written to a commandable object

Operations available on one object 'using' a given value:

set	Set object to given value (unsigned 32-bit integer)
skip = N	Skip next operation if object is equal to given value
skip < N	Skip next operation if object is less than given value
skip > N	Skip next operation if object is greater than given value
comp = N	Compare, set result 'true' if object is equal to given value
comp < N	Compare, set result 'true' if object is less than given value
comp > N	Compare, set result 'true' if object is greater than given value
pack	Perform Pack operation (see text)
fill	Perform Fill operation (see text)

unpack	Perform Unpack operation (see text)
priority	Sets command priority that will be used in any subsequent write to a commandable object

Operations "using" a given value will have an unsigned integer value in the "This Object/Value" column rather than an object number. These values will be displayed as integer for most operations, but will be displayed in hexadecimal for pack, fill, and unpack operations since these operate primarily on bit mask values.

The result of a test or compare will be zero if false, or one if true when the result object is a Analog or Binary object. The result of a test or compare when the result object is Multi-State will be 1 if false and 2 if true (since Multi-State cannot use zero).

The next two screen shots illustrate compare, set, and skip operations. Rule 5 says that rule 6 will not be executed if AI 6 contains a zero. If AI 6 is not equal to zero, then rule 6 will be executed. (The numbers rule 6 and AI 6 are not related in any other way, this is just coincidence in the example.)

Calculate		Copy		Report			
Showing 1 to 8 of 8						Update < Prev Next >	
Rule #	Perform Operation	Using Object	And/Thru Using	This Object/Value	Place Result in Object		
1	comp = N	AI 1	using	10	AI 2		
2	comp < N	AI 1	using	10	AI 3		
3	comp > N	AI 1	using	10	AI 4		
4	set	AI 5	using	202	AI 5		
5	skip = N	AI 6	using	0	AI 6		
6	set	AI 7	using	0	AI 7		
7	set	AI 8	using	88	AI 8		
8	none	None	and	None	None		
# Rules Enabled: 8						Insert Delete	

Object values for examples using the above operations are illustrated below.

Local Objects		BACnet	Modbus	System		
Analog		Binary	Multi-State	Actions		
Input Objects		Output Objects	Value Objects			
Analog Input Objects		Showing objects from 1		Refresh	< Prev	Next >
Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Units
<u>1</u>	Analog Input 1 Description of AI 1	N	10.000000	0	0,0,0,0	no_units
<u>2</u>	Analog Input 2 Description of AI 2	N	1.000000	0	0,0,0,0	no_units
<u>3</u>	Analog Input 3 Description of AI 3	N	0.00	0	0,0,0,0	no_units
<u>4</u>	Analog Input 4 Description of AI 4	N	0.00	0	0,0,0,0	no_units
<u>5</u>	Analog Input 5 Description of AI 5	N	202.0000	0	0,0,0,0	no_units
<u>6</u>	Analog Input 6 Description of AI 6	N	1.000000	0	0,0,0,0	no_units
<u>7</u>	Analog Input 7 Description of AI 7	N	0.00	0	0,0,0,0	no_units
<u>8</u>	Analog Input 8 Description of AI 8	N	88.000000	0	0,0,0,0	no_units

The following screen shot illustrates the use of calculate rules to set the states of multiple Binary Input objects based on the value of a single Multi-State Input object. In this example, BI 1 will be active if the MI 1 state is 1, BI 2 will be active for state 2, and so on.

Calculate		Copy	Report			
Showing 1		to 5 of 5		Update	< Prev	Next >
Rule #	Perform Operation	Using Object	And/Thru Using	This Object/Value	Place Result in Object	
1	comp = N	MI 1	using	1	BI 1	
2	comp = N	MI 1	using	2	BI 2	
3	comp = N	MI 1	using	3	BI 3	
4	comp = N	MI 1	using	4	BI 4	
5	none	None	and	None	None	
# Rules Enabled: 5		Insert		Delete		

The calculate rules have access to command priority and relinquish when the result register is a commandable object.

The command priority is set using the priority operation as illustrated below. In this case, the "using object" and "place result in object" are only place holders to keep the rule validator happy. The only thing actually used in this operation is the "this value". In the example below, the command priority is being set to 7. This command priority will be used for any subsequent operations that place a result in a commandable object, and will remain in effect until another priority operation is used. If no priority operation is ever included, then the default local command priority on the BACnet settings page will be used.

Once a commandable object has been set by a calculate rule, it can be relinquished by using the relinquish operation as illustrated below. The command priority currently in effect as the result of the most recent priority operation will be relinquished. The

calculate rules themselves do not have any ability to remember command priorities - it is up to you to keep track of command priority using the priority operation.

Calculate		Copy		Report			
Showing 1 to 6 of 6						Update	< Prev Next >
Rule #	Perform Operation	Using Object	And/Thru Using	This Object/Value	Place Result in Object		
1	priority	AI 1	using	7	AI 1		
2	set	AO 1	using	55	AO 1		
3	skip = N	AI 2	using	0	AI 2		
4	relinquish	AO 1	and	None	AO 1		
5	set	AI 3	using	15	AI 3		
6	none	None	and	None	None		
# Rules Enabled: 6						Insert	Delete

Pack and fill are used for packing multiple local objects into a single object for purposes of emulating existing equipment when the Babel Buster is functioning as a server (slave). When pack and fill are used, "using" should be selected, and the second entry is a hexadecimal mask or fill value. The hexadecimal value should include "h" at the end to signify hexadecimal (otherwise the value will be parsed as decimal).

The pack mask is both a bit mask and position indicator. To calculate the contribution of a given calculate rule, the mask is right shifted until the least significant bit is nonzero, then this shifted mask is logically AND-ed with the local object content. The resulting masked value is then left shifted back to the original mask position. This final shifted result is then logically OR-ed into the result object (after first clearing the bits in the affected position of the result object).

Fill is simple - it simply logically OR's the bit mask into the result object.

Calculate		Copy		Report			
Showing 1 to 4 of 4						Update	< Prev Next >
Rule #	Perform Operation	Using Object	And/Thru Using	This Object/Value	Place Result in Object		
1	pack	AI 2	using	Fh	AI 1		
2	pack	AI 3	using	F0h	AI 1		
3	pack	AI 4	using	F00h	AI 1		
4	none	None	and	None	None		
# Rules Enabled: 4						Insert	Delete

Using the above rules, an example of resulting data would be as follows.

Local Objects	BACnet	Modbus	System
Analog	Binary	Multi-State	Actions
Input Objects	Output Objects	Value Objects	

Analog Input Objects Showing objects from 1 Refresh < Prev Next >

Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Units
1	Analog Input 1 Description of AI 1	N	273.0000	0	0,0,0,0	no_units
2	Analog Input 2 Description of AI 2	N	1.000000	0	0,0,0,0	no_units
3	Analog Input 3 Description of AI 3	N	1.000000	0	0,0,0,0	no_units
4	Analog Input 4 Description of AI 4	N	1.000000	0	0,0,0,0	no_units
5	Analog Input 5 Description of AI 5	N	0.00	0	0,0,0,0	no_units

A set of calculate rules that would exactly reverse the above operation would be as follows.

Calculate	Copy	Report
-----------	------	--------

Showing 1 to 4 of 4 Update < Prev Next >

Rule #	Perform Operation	Using Object	And/Thru Using	This Object/Value	Place Result in Object
1	unpack ▼	AI 1	using ▼	Fh	AI 2
2	unpack ▼	AI 1	using ▼	F0h	AI 3
3	unpack ▼	AI 1	using ▼	F00h	AI 4
4	none ▼	None	and ▼	None	None

Rules Enabled: 4 Insert Delete

The pack, fill, and unpack instructions are primarily targeting Modbus applications. They are less useful when dealing with BACnet objects, but are retained in the calculate rule set for consistency across the gateway family.

6.5 Local Object Copy Rules

The copy rules provide a means of simply copying the content of one object to another.

Babel Buster[®] 3
BACNET
ROUTER & GATEWAY
MODEL BB3-7302-V2C

CONTROL SOLUTIONS MINNESOTA

Local Objects BACnet SNMP System

Analog Binary Multi-State Actions

Calculate Copy Report

Showing 1 to 2 of 2 Update < Prev Next >

Rule #	Source Object		Destination Object	
1	AI 7		AI 14	
2	None		None	

Rules Enabled: 2 Insert Delete

The above rule would cause the value of AI 7 to be placed in AI 14. If a floating point (Analog) value is copied to a Binary object, the Binary object will be set Active if the value was nonzero, or cleared to Inactive if zero. Analog values copied to a Multistate object will be not only truncated, but bounded to the maximum number of states (not a recommended use of Copy).

6.6 Device Status Reporting

The Babel Buster BB3-7302/MX-73 read maps include the ability to set a default value upon 'n' read fails, meaning that if the Babel Buster gets an error 'n' times attempting to read that point, it will automatically set the corresponding local object to the given default value to indicate the problem. This indication applies on a point by point basis, but of course any one point can be used as an indication that the entire device may be offline.

The BB3-7302/MX-73 also includes the ability to report device errors to an assigned status object rather than rely on default values. This reporting is configured on the Report page.

Babel Buster[®] 3
BACNET
ROUTER & GATEWAY
MODEL BB3-7302-V2C

CONTROL SOLUTIONS MINNESOTA

Local Objects BACnet SNMP System

Analog Binary Multi-State Actions

Calculate Copy **Report**

Showing 1 to 7 of 7 Update < Prev Next >

Report Status of	Device or Unit #	To This Object	With This Delay (Sec.)	Delete
BACnet IP Client	1	AV 1	20	<input type="checkbox"/>
BACnet IP Client	2	AV 2	20	<input type="checkbox"/>

BACnet IP Client 1 AI 1 20 Add

This optional list allows reporting device errors as object values to make it easier to monitor communication failures. The length of the list is variable. To add to the list, select the type of device to report on, select the device instance or unit number to report on, and select an object in which to put the status indication. Enter a delay if desired, and then click Add.

The delay is technically optional (zero means no delay) but you will generally want to set the delay time greater than the read/write map poll times. If some number of seconds is entered, then the error condition will not be reported until this time period expires. If the error clears before the time is up, then the error is never reported. This is useful for spurious errors that would result in nuisance indications.

To remove a report from the list, check the box in the Delete column and then click Update. Click Prev or Next to scroll through the list.

Error codes placed into the reporting object will be as follows:

- 0 = No error
- 1 = Timeout, no response from remote device
- 2 = Error message received from remote device
- 3 = Line fault (e.g. CRC error, socket connection error, etc)

Once a Timeout error indication has been set (following delay if applicable), it will automatically return to zero upon the next successful communication with that device.

Once either the error message or line fault indication has been set, following delay if applicable, communication must continue free of this same error condition for at least the same delay period before the indication will be reset to zero. If an error is reported for one data point, but multiple others are error free, then the one error would be hidden without this delay before reset. Ideally, this delay period should be at least as great as the poll period for the slowest point mapped.



7. Configuring Gateway as a BACnet Client

The BACnet client is used to query other BACnet devices, obtain their Present Value data (or other property value), and store a copy of that data in the BB3-7302/MX-73's own local objects. From there, the data may be accessed by SNMP devices or other BACnet devices.

This data exchange with other BACnet devices requires that you define those devices in the list on the BACnet Client Devices page, and then create some number of Read and Write Maps. The maps may be created via the web pages talked about below. But you also have the option of using a standard spread sheet program to create a list that you save as a CSV file, and then import that via the File Manager in the BB3-7302/MX-73. Refer to Section 3.1 for more about importing CSV files, and also Appendix C for CSV file format information.

7.1 BACnet Device List

Setting up the BACnet client consists of identifying one or more BACnet devices, then listing the objects that should be queried (whether read or written). The client configuration pages are illustrated below.

Babel Buster[®] 3
BACNET
ROUTER & GATEWAY
MODEL BB3-7302-V2C

CONTROL SOLUTIONS MINNESOTA

Local Objects | BACnet | SNMP | System

Local Device | BACnet Client | Diagnostics | BBMD

Devices | Client Read Map | Client Write Map

Device # 1 [Update] [< Prev] [Next >]

Device Instance: 64 Local Name: BACnet Test Server 1

Default Poll Period: 5.0 Seconds Default Write Priority: 10

Reply Timeout: 2.0 Seconds Timeouts: 0 [Clear]

Address Binding: ☒ Dynamic (Who-Is) ☐ Static

Device Address: 192.168.1.64:47808 Net 8 [Clear Cache]

Network Number: 0 MAC Length: 0

MAC Address: 0

Device number simply shows you where you are on the internal local device list. Click "next" and "prev" to scroll through the list.

Remote BACnet devices to be accessed by this device are specified here. Enter the Device Instance of the remote device, a name to reference in other pages, and a poll rate. Then click "Update".

Select dynamic or static address binding. Dynamic binding is used most often, and simply means the gateway will send out a "Who-Is" request asking for the device instance to respond, at which time the gateway learns its IP or MS/TP address automatically.

When dynamic address binding is used (default), the gateway broadcasts a "Who-Is" looking for this device instance when a read or write map wants to use this device. When (if) it responds, its IP address is listed here simply as a diagnostic. Timeouts resulting from inability to reach this device are tabulated on this page as well, and may be cleared by clicking the Clear button. To cause the who-is process to be repeated, click Clear Cache. When dynamic binding is used, the IP address is read-only and any changes entered will be ignored.

The screenshot shows the 'BACnet Client' configuration window for Device # 5. The interface has a top navigation bar with 'Local Device', 'BACnet Client', 'Diagnostics', and 'BBMD'. Below this is a sub-navigation bar with 'Devices', 'Client Read Map', and 'Client Write Map'. The 'Devices' section is active, showing fields for Device # 5, Device Instance 20822, Local Name 'BACnet Test Server 5', Default Poll Period 5.0 Seconds, Default Write Priority 10, Reply Timeout 2.0 Seconds, Timeouts 0, Address Binding set to Dynamic (Who-Is), Device Address 'MS/TP 22 Net 115', Network Number 0, MAC Length 0, and MAC Address 0. There are 'Update', '< Prev', and 'Next >' buttons at the top right, and 'Clear' and 'Clear Cache' buttons on the right side of the form.

If static binding must be used, enter the fixed IP address you know the device instance to be found at. If no port is given, it will default to 0xBAC0 (47808). Enter IP as a.b.c.d or IP with port as a.b.c.d:p, for example 192.168.1.99:47808. Network number 0 means local IP network. If the device is on the other side of a router, the MAC Address entry must be used instead.

The screenshot shows the 'BACnet Client' configuration window for Device # 1. The interface has a top navigation bar with 'Local Device', 'BACnet Client', 'Diagnostics', and 'BBMD'. Below this is a sub-navigation bar with 'Devices', 'Client Read Map', and 'Client Write Map'. The 'Devices' section is active, showing fields for Device # 1, Device Instance 64, Local Name 'BB2-7010-01', Default Poll Period 5.0 Seconds, Default Write Priority 10, Reply Timeout 1.0 Seconds, Timeouts 0, Address Binding set to Static, Device Address '192.168.1.64:47808', Network Number 0, MAC Length 0, and MAC Address. There are 'Update', '< Prev', and 'Next >' buttons at the top right, and 'Clear' and 'Clear Cache' buttons on the right side of the form.

If the device is on the local MS/TP link then enter its MS/TP MAC address as the device address using "MS/TP" followed by the address. Network number 0 means local MS/TP network. If the device is on the other side of a router, the MAC Address entry must be used instead.

The screenshot shows the 'BACnet Client' configuration window for Device # 5. The interface has a top navigation bar with 'Local Device', 'BACnet Client', 'Diagnostics', and 'BBMD'. Below this is a sub-navigation bar with 'Devices', 'Client Read Map', and 'Client Write Map'. The 'Devices' tab is active, showing a list of devices with 'Device # 5' selected. The configuration fields for Device # 5 are as follows:

Field	Value
Device Instance	20822
Local Name	BACnet Test Server 5
Default Poll Period	5.0 Seconds
Default Write Priority	10
Reply Timeout	2.0 Seconds
Timeouts	0
Address Binding	<input checked="" type="radio"/> Dynamic (Who-Is) <input type="radio"/> Static
Device Address	MS/TP 22
Network Number	0
MAC Length	0
MAC Address	

Buttons: Update, < Prev, Next >, Clear, Clear Cache.

Include network number, mac length, and mac address ONLY if static binding to a device on the other side of some other BACnet router. In the example below, the device is an MS/TP device at MAC address 22 on network 115 on the other side of a router at 192.168.1.115 which is connected to the local IP network.

The screenshot shows the 'BACnet Client' configuration window for Device # 1. The interface is the same as the previous one, but the configuration fields for Device # 1 are as follows:

Field	Value
Device Instance	20822
Local Name	Remote Server
Default Poll Period	5.0 Seconds
Default Write Priority	10
Reply Timeout	2.0 Seconds
Timeouts	0
Address Binding	<input checked="" type="radio"/> Dynamic (Who-Is) <input type="radio"/> Static
Device Address	192.168.1.115:47808
Network Number	115
MAC Length	1
MAC Address	22

Buttons: Update, < Prev, Next >, Clear, Clear Cache.

In the example below, the device is a BACnet IP device at IP address 192.168.1.77 (port 47808) on network 88 on the other side of a router at MAC address 91 which is connected to the local MS/TP network.

When using static binding to a device on the other side of another router, the Device Address must always be the address of that other router and the Device Address given must be connected to the local network.

Local Device | BACnet Client | Diagnostics | BBMD

Devices | Client Read Map | Client Write Map

Device # 1 [Update] [< Prev] [Next >]

Device Instance 77 Local Name: Server 77

Default Poll Period 5.0 Seconds Default Write Priority: 10

Reply Timeout: 2.0 Seconds Timeouts: 0 [Clear]

Address Binding: ☒ Dynamic (Who-Is) ☐ Static

Device Address MS/TP 91 [Clear Cache]

Network Number 88 MAC Length 6

MAC Address C0:A8:01:4D:BA:C0

7.2 BACnet Client Read Maps

Getting the gateway to read objects from another BACnet device requires setting up a "Read Map" as shown here.

Babel Buster[®] 3
BACNET
ROUTER & GATEWAY
MODEL BB3-7302-V2C

CONTROL SOLUTIONS MINNESOTA

Local Objects | BACnet | SNMP | System

Local Device | BACnet Client | Diagnostics | BBMD

Devices | Client Read Map | Client Write Map

Showing 1 to 11 of 11 [Update] [< Prev] [Next >]

Map #	Remote Type	Remote Object #	Remote Device	Scale	Local Object #	Name
1	Analog Input	1	BACnet Test Server	0.00	AI 1	Analog Input 1
2	Analog Output	1	BACnet Test Server	0.00	AO 1	Analog Output 1
3	Analog Value	1	BACnet Test Server	1.800000	AV 1	Analog Value 1
4	Binary Input	1	BACnet Test Server	0.00	BI 1	Binary Input 1
5	Binary Output	1	BACnet Test Server	0.00	BO 1	Binary Output 1
6	Binary Value	1	BACnet Test Server	0.00	BV 1	Binary Value 1
7	Multistate Input	1	BACnet Test Server	0.00	MI 1	Multi-state Input 1
8	Multistate Output	1	BACnet Test Server	0.00	MO 1	Multi-state Output 1
9	Multistate Value	1	BACnet Test Server	0.00	MV 1	Multi-state Value 1
10	Analog Output	1	BACnet Test Server	0.00	AI 3	Analog Input 3
11	None	0	None	0.00	None	---

Map number simply tells you where you're at on the list of object maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Showing" box, then click Update.

Maps entered on this page only read data from remote devices. Go to the Client Write Map to write data to those devices. The full parameter set is different for read versus write.

An abbreviated version of a list of maps is shown on this page. Any of the parameters shown may be changed here and registered by clicking the Update button. To view and/or modify the complete set of parameters, click on the map number in the left most column.

For each remote object to be read, select the object instance and type, and remote device. The names in the remote device list are defined in the Devices page. The property read will default to Present Value. If you wish to read a different property, click on the Map # in the first column for the expanded view of the map and enter the property number.

When the remote object is read, data may be manipulated before being written to the local object. The value will be multiplied by the scale factor. The final result is written to the local object number given. The name is optional and used only for display purposes.

Selecting "none" for remote type effectively deletes the map even though it will still appear in the list until deleted. Unused maps at the end of the list will always show none as the type.

Local Object is internally a coded number consisting of BACnet object type multiplied by 10000, then added to the object number starting from #1. These numbers will appear as "register numbers" in XML configuration files. These are translated into abbreviations that are easy to interpret on the web page as follows:

AI n = Analog Input #n
AO n = Analog Output #n
AV n = Analog Value #n
BI n = Binary Input #n
BO n = Binary Output #n
BV n = Binary Value #n
MI n = Multi-state Input #n
MO n = Multi-state Output #n
MV n = Multi-state Value #n

Local object numbers start at #1. The maximum available number varies by object type, and these limits are set on the Resources page (under System).

Click on a Map # in the first column of maps to get the expanded view of that map as follows:

The screenshot shows a software interface for configuring a BACnet Client. At the top, there are tabs for 'Local Objects', 'BACnet', 'SNMP', and 'System'. Under the 'BACnet' tab, there are sub-tabs for 'Local Device', 'BACnet Client', 'Diagnostics', and 'BBMD'. The 'BACnet Client' sub-tab is active, showing 'Client Read Map' and 'Client Write Map' options. The 'Client Read Map' option is selected, and 'Map # 1' is entered in the 'Map #' box. The configuration details for Map #1 are as follows:

- Read property:** Present Value (dropdown), 85 (text box), from instance # 1 (text box) of object type Analog Input (dropdown).
- Read from device:** BACnet Test Server (dropdown), using index ALL (text box).
- Then apply scale:** 0.00 (text box) and offset: 0.00 (text box).
- Save in local object:** AI 1 (text box) named Analog Input 1 (text box). Repeat this process every 5.0 (text box) seconds.
- Apply this default value:** 120.0000 (text box) after 3 (text box) read failure(s).
- ☐ Enable this map only when index object None (text box) is set to a value of 0 (text box).

At the bottom, it shows '# Client Read Maps Enabled: 11' and buttons for 'Insert' and 'Delete'.

Map number simply tells you where you're at on the list of object maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Map #" box, then click Update.

For each remote object to be read, enter the property number, object instance and type, and select a remote device. The names in the device list are defined in the Devices page. Use index value of "ALL" if no index (use this by default if you do not have a known index number).

The most commonly read property will be Present Value, which is property number 85. For other property numbers, refer to Appendix E, BACnet Codes.

When the remote object is read, data may be manipulated before being written to the local object. The value will be multiplied by the scale factor, then the offset is added. The final result is written to the local object number given. The name is optional and used only for display purposes.

The periodic poll time ("Repeat this process") determines how often the remote object will be read. This number, if nonzero, will override the default poll time given in the Devices page for the remote device being read.

The default value will be stored into the local object after the given number of read failures if the fail count is non-zero. Setting the count to zero will disable the default, and the object will retain the most recent value obtained.

You have the option of enabling this map only when a selected object contains a given value. Any local object may be used as the index object. As the name implies, you could have the same local object contain different values based on different maps as indexed by the index object.

Delete will remove the map number shown in the "Map #" box. Insert will insert a new map before the map number shown, and is used for placing maps between existing maps. It is not necessary to use Insert to add maps to the bottom of the list or to

define any map presently having zero for a source object or "none" for remote type.

Selecting "none" for remote type effectively deletes the map even though it will still appear in the list until deleted. Unused maps at the end of the list will always show none as the type. If you wish to prevent these from being displayed, reduce the number of maps enabled.

The number of maps enabled simply limits the scope of map review so that you do not have to review a lot of unused maps. If the displayed maps are used up and you need more, increase the enabled number.

The expanded view of the Client Read Map may look daunting at first. Here is the same screen shot with the parts you are most likely to not use grayed out for illustration purposes. If you are only concerned with reading Present Value and you have set the default poll time on the Device page, then you really never need to look at the expanded view at all. Your configuration can be entered entirely on the tabular list of maps. The same applies to Write Maps below.

The screenshot shows the 'Client Read Map' configuration window. At the top, there are tabs for 'Devices', 'Client Read Map', and 'Client Write Map'. Below the tabs, there is a 'Map #' field set to 1, and buttons for 'Update', '< Prev', and 'Next >'. The main configuration area includes several fields: 'Read property' set to 'Present Value', a value of 85, 'from instance #' set to 1, and 'of object type' set to 'Analog Input'. Below this, 'Read from device' is set to 'BACnet Test Server' and 'using index' is set to 'ALL'. Further down, 'Then apply scale:' is set to 0.00 and 'and offset:' is set to 0.00. The 'Save in local object' field is set to 'AI 1', named 'Analog Input 1', and 'Repeat this process every' is set to 5.0 seconds. At the bottom, 'Apply this default value:' is set to 120.0000, 'after' 3 read failure(s). There is also a checkbox for 'Enable this map only when index object' set to 'None' is set to a value of 0. At the very bottom, there is a field for '# Client Read Maps Enabled' set to 11, and buttons for 'Insert' and 'Delete'.

7.3 BACnet Client Write Maps

Getting the gateway to write objects to another BACnet device requires setting up a "Write Map" as shown here. Much of the Write Map is configured the same as a Read Map.

Babel Buster[®] 3
BACNET
ROUTER & GATEWAY
MODEL BB3-7302-V2C

CONTROL SOLUTIONS MINNESOTA

Local Objects BACnet SNMP System

Local Device BACnet Client Diagnostics BBMD

Devices Client Read Map **Client Write Map**

Showing to 9 of 9

Map #	Local Object #	Scale	Remote Type	Remote Object #	Remote Device	Name
1	AO 2	0.00	Analog Output ▾	2	BACnet Test Server ▾	Analog Output 2
2	AV 2	0.555550	Analog Value ▾	2	BACnet Test Server ▾	Analog Value 2
3	BO 2	0.00	Binary Output ▾	2	BACnet Test Server ▾	Binary Output 2
4	BV 2	0.00	Binary Value ▾	2	BACnet Test Server ▾	Binary Value 2
5	MO 2	0.00	Multistate Output ▾	2	BACnet Test Server ▾	Multi-state Output 2
6	MV 2	0.00	Multistate Value ▾	2	BACnet Test Server ▾	Multi-state Value 2
7	AV 3	0.00	Analog Value ▾	3	BACnet Test Server ▾	Analog Value 3
8	AV 4	0.00	Analog Value ▾	4	BACnet Test Server ▾	Analog Value 4
9	None	0.00	None ▾	0	None ▾	---

Map number simply tells you where you're at on the list of object maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Showing" box, then click Update.

Maps entered on this page only write data to remote devices. Go to the Client Read Map to read data from those devices. The full parameter set is different for read versus write.

An abbreviated version of a list of maps is shown on this page. Any of the parameters shown may be changed here and registered by clicking the Update button. To view and/or modify the complete set of parameters, click on the map number in the left most column.

For each remote object to be written, select the object instance and type, and remote device. The names in the device list are defined in the Devices page. The property written will default to Present Value. If you wish to write a different property, click on the Map # in the first column for the expanded view of the map and enter the property number.

Data from the local object given will be multiplied by the scale factor before being written.

Selecting "none" for remote type effectively deletes the map even though it will still appear in the list until deleted. Unused maps at the end of the list will always show none as the type.

Local Object is internally a coded number consisting of BACnet object type multiplied by 10000, then added to the object number starting from #1. These numbers will appear as "register numbers" in XML configuration files. These are translated into abbreviations that are easy to interpret on the web page as follows:

AI n = Analog Input #n
 AO n = Analog Output #n
 AV n = Analog Value #n
 BI n = Binary Input #n
 BO n = Binary Output #n
 BV n = Binary Value #n
 MI n = Multi-state Input #n
 MO n = Multi-state Output #n
 MV n = Multi-state Value #n

Object numbers start at #1. The maximum available number varies by object type, and these limits are set on the Resources page (under System).

Click on a Map # in the first column of maps to get the expanded view of that map as follows:

The screenshot shows a web-based configuration interface for a BACnet Client. The top navigation bar includes tabs for Local Objects, BACnet, SNMP, and System. The BACnet tab is active, showing sub-tabs for Local Device, BACnet Client, Diagnostics, and BBMD. The 'Client Write Map' sub-tab is selected. Below the navigation bar, there is a 'Map #' field with the value '1' and buttons for 'Update', '< Prev', and 'Next >'. The main configuration area for Map #1 is titled 'Read local object AO 2 named Analog Output 2'. It contains several fields and checkboxes: 'Write remote object' with a checked box and two options (change by 0.00 or when 0.0 seconds have elapsed), 'Apply scale: 0.00 and offset: 0.00', 'Then, using index ALL and priority 8 proceed to', 'Write property Present Value 85 encoded as data type Real', 'Write to instance #2 of object type Analog Output at device BACnet Test Server', 'Repeat this process at least or no more than every 0.0 seconds', and a checked box 'Enable this map only when index object BI 3 is set to a value of 1'. At the bottom, there is a '# Client Write Maps Enabled: 9' field and 'Insert' and 'Delete' buttons.

Map number simply tells you where you're at on the list of object maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Map #" box, then click Update.

The local object data may be written periodically, or when it changes, or both. To send upon change (send on delta), check the first box and enter the amount by which the local object must change before being written to the remote device. To guarantee that the remote object will be written at least occasionally even if the data does not change, check the second box and enter some amount of time. This time period will be

referred to as the "maximum quiet time".

Data from the local object may be manipulated before being written to the remote object. The local data is first multiplied by the scale factor. The offset is then added to it.

For the remote object to be written, enter the property number, object instance and type, index if applicable (enter "ALL" if no index), and priority to use of the object being written is commandable. Select a remote device to write to. The names in the device list are defined in the Devices page.

The most commonly written property will be Present Value, which is property number 85. For other property numbers, refer to Appendix E, BACnet Codes.

The repeat time may determine how often the remote object will be written. If send on delta and maximum quiet time are not checked above, clicking the "at least" button will establish a periodic update time. If send on delta is used and you wish to limit the network traffic in the event changes are frequent, click the "no more than" button and enter the minimum time that should elapse before another write to the remote device.

You have the option of enabling this map only when a selected object contains a given value. Any local object may be used as the index object. As the name implies, you can write different values to the remote object based on different maps as indexed by the index object.

Delete will remove the map number shown in the "Map #" box. Insert will insert a new map before the map number shown, and is used for placing maps between existing maps. It is not necessary to use Insert to add maps to the bottom of the list or to define any map presently having zero for a source object or "none" for remote type.

Selecting "none" for remote type effectively deletes the map even though it will still appear in the list until deleted. Unused maps at the end of the list will always show none as the type. If you wish to prevent these from being displayed, reduce the number of maps enabled.

The number of maps enabled simply limits the scope of map review so that you do not have to review a lot of unused maps. If the displayed maps are used up and you need more, increase the enabled number.

7.4 BACnet Client Diagnostics

If errors are detected in the course of reading or writing other BACnet objects via the client's maps, they will be indicated on the errors pages.

Local Objects	BACnet	SNMP	System		
	Local Device	BACnet Client	Diagnostics	BBMD	
Errors: Read Maps	Errors: Write Maps				

Map #	Remote Type	Remote Object #	Remote Device	Name	Error Class	Error Code
3	AV	6125	BACnet Test Server	Analog Value 1	1	31

Errors for BACnet client read/write maps are shown on these pages. Only those maps with errors to report are listed. Refer to the code and class lists below for interpretation. In the illustration above, error class 1 says the error refers to "object" and the code says "unknown object". In other words, AV 6125 does not exist in the device shown.

Proprietary class 82, code 0, is generated locally indicating a timeout, no response received from remote server. All other codes listed below are returned by the remote server.

```
0 = ERROR_CLASS_DEVICE
1 = ERROR_CLASS_OBJECT
2 = ERROR_CLASS_PROPERTY
3 = ERROR_CLASS_RESOURCES
4 = ERROR_CLASS_SECURITY
5 = ERROR_CLASS_SERVICES
```

```
/* valid for all classes */
0 = ERROR CODE OTHER
```

```
/* Error Class - Device */
2 = ERROR_CODE_CONFIGURATION_IN_PROGRESS
3 = ERROR_CODE_DEVICE_BUSY
25 = ERROR_CODE_OPERATIONAL_PROBLEM
```

```
/* Error Class - Object */
4 = ERROR_CODE_DYNAMIC_CREATION_NOT_SUPPORTED
17 = ERROR_CODE_NO_OBJECTS_OF_SPECIFIED_TYPE
23 = ERROR_CODE_OBJECT_DELETION_NOT_PERMITTED
24 = ERROR_CODE_OBJECT_IDENTIFIER_ALREADY_EXISTS
27 = ERROR_CODE_READ_ACCESS_DENIED
31 = ERROR_CODE_UNKNOWN_OBJECT
36 = ERROR_CODE_UNSUPPORTED_OBJECT_TYPE
```

```
/* Error Class - Property */
8 = ERROR_CODE_INCONSISTENT_SELECTION_CRITERION
9 = ERROR_CODE_INVALID_DATA_TYPE
32 = ERROR_CODE_UNKNOWN_PROPERTY
```

```
37 = ERROR_CODE_VALUE_OUT_OF_RANGE
40 = ERROR_CODE_WRITE_ACCESS_DENIED
41 = ERROR_CODE_CHARACTER_SET_NOT_SUPPORTED
42 = ERROR_CODE_INVALID_ARRAY_INDEX
44 = ERROR_CODE_NOT_COV_PROPERTY
45 = ERROR_CODE_OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED
47 = ERROR_CODE_DATATYPE_NOT_SUPPORTED
50 = ERROR_CODE_PROPERTY_IS_NOT_AN_ARRAY
```

```
/* Error Class - Resources */
```

```
18 = ERROR_CODE_NO_SPACE_FOR_OBJECT
19 = ERROR_CODE_NO_SPACE_TO_ADD_LIST_ELEMENT
20 = ERROR_CODE_NO_SPACE_TO_WRITE_PROPERTY
```

```
/* Error Class - Security */
```

```
1 = ERROR_CODE_AUTHENTICATION_FAILED
6 = ERROR_CODE_INCOMPATIBLE_SECURITY_LEVELS
12 = ERROR_CODE_INVALID_OPERATOR_NAME
15 = ERROR_CODE_KEY_GENERATION_ERROR
26 = ERROR_CODE_PASSWORD_FAILURE
28 = ERROR_CODE_SECURITY_NOT_SUPPORTED
30 = ERROR_CODE_TIMEOUT
```

```
/* Error Class - Services */
```

```
5 = ERROR_CODE_FILE_ACCESS_DENIED
7 = ERROR_CODE_INCONSISTENT_PARAMETERS
10 = ERROR_CODE_INVALID_FILE_ACCESS_METHOD
11 = ERROR_CODE_ERROR_CODE_INVALID_FILE_START_POSITION
13 = ERROR_CODE_INVALID_PARAMETER_DATA_TYPE
14 = ERROR_CODE_INVALID_TIME_STAMP
16 = ERROR_CODE_MISSING_REQUIRED_PARAMETER
22 = ERROR_CODE_PROPERTY_IS_NOT_A_LIST
29 = ERROR_CODE_SERVICE_REQUEST_DENIED
43 = ERROR_CODE_COV_SUBSCRIPTION_FAILED
46 = ERROR_CODE_INVALID_CONFIGURATION_DATA
48 = ERROR_CODE_DUPLICATE_NAME
49 = ERROR_CODE_DUPLICATE_OBJECT_ID
```

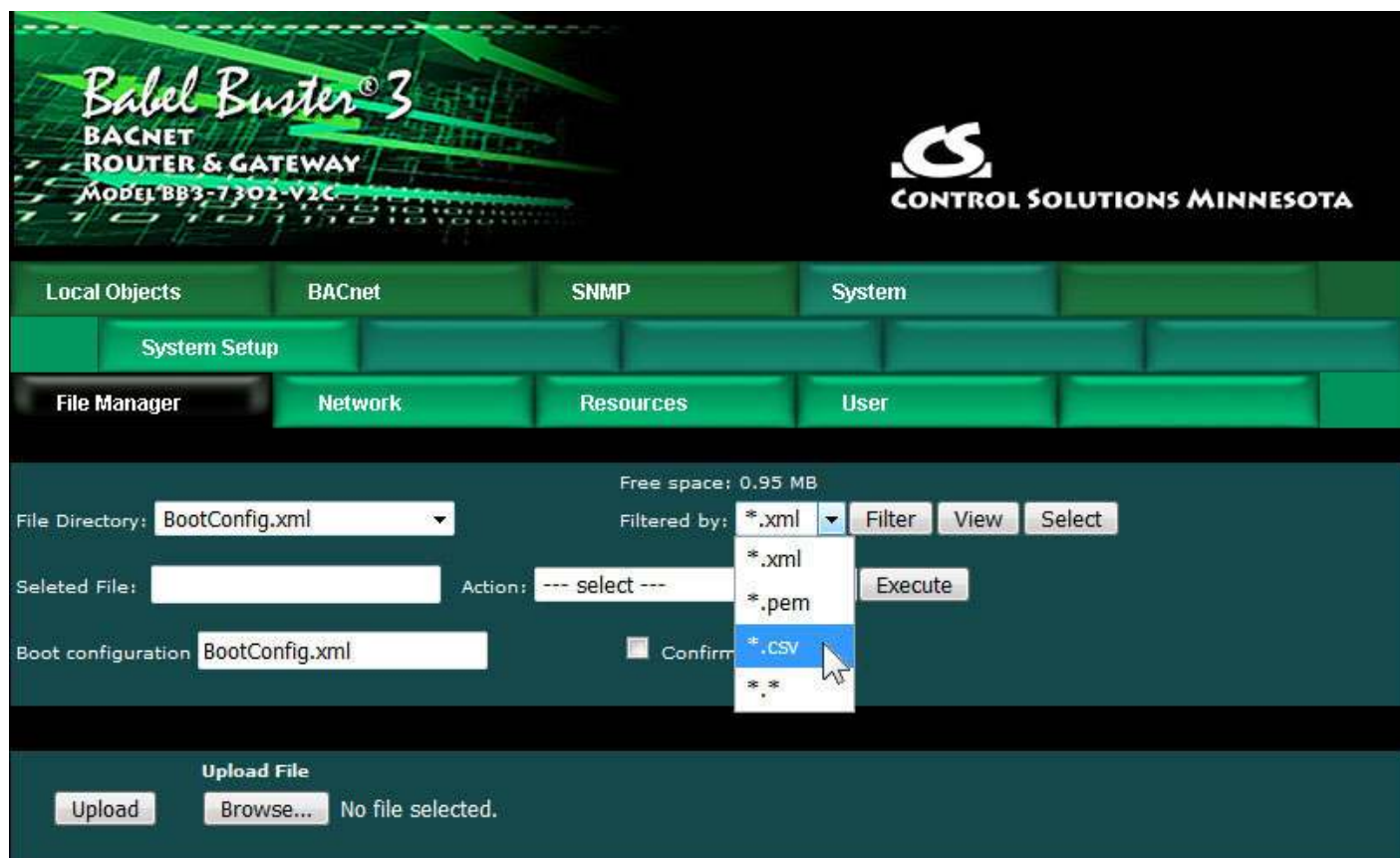
7.5 Importing BACnet Client Maps from CSV File

The built-in web user interface is user friendly, but can get tedious if you have a lot of maps to enter. You may already have a list of BACnet objects available in spread sheet form. With a bit of editing, you can turn this into a CSV file that can be directly imported into the BB3-7302/MX-73 to quickly configure a lot of read and write maps. If you are proficient with spread sheets, you can probably create a rather large set of maps quickly and speed up the process of configuring the BB3-7302/MX-73.

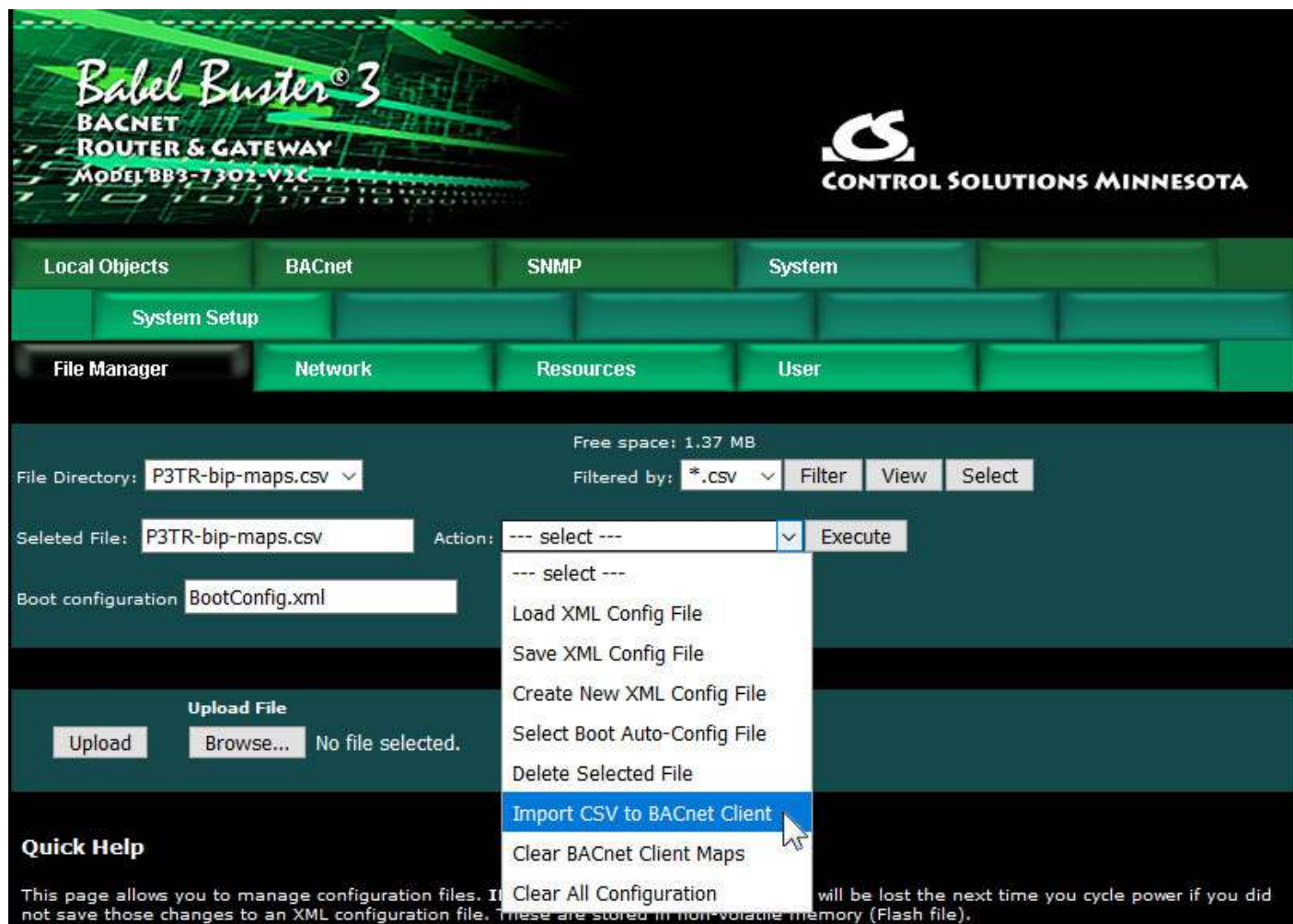
There is more discussion about the File Manager in Section 3, but a summary of what you need to do to import maps from a CSV file is given here.

Start by uploading your CSV file. Use the Browse button to locate the file on your PC, then click Upload.

Select *.csv as the file filter. This will result in showing the list of CSV files currently stored in the BB3-7302/MX-73.



Select your file from the File Directory drop-down list, then click the Select button on the right. Select "Import CSV to BACnet Client" from the Action list, and click Execute.



We imported 220 read maps in our example test case.

Local Objects		BACnet		SNMP		System	
Local Device		BACnet Client		Diagnostics		BBMD	
Devices		Client Read Map		Client Write Map			
Showing 1 to 15 of 221							
				Update		< Prev Next >	
Map #	Remote Type	Remote Object #	Remote Device	Scale	Local Object #	Name	
1	Analog Input ▾	1	BACnet Test Server ▾	0.00	AI 1	Analog Input 1	
2	Analog Input ▾	2	BACnet Test Server ▾	0.00	AI 2	Analog Input 2	
3	Analog Input ▾	3	BACnet Test Server ▾	0.00	AI 3	Analog Input 3	
4	Analog Input ▾	4	BACnet Test Server ▾	0.00	AI 4	Analog Input 4	
5	Analog Input ▾	5	BACnet Test Server ▾	0.00	AI 5	Analog Input 5	
6	Analog Input ▾	6	BACnet Test Server ▾	0.00	AI 6	Analog Input 6	
7	Analog Input ▾	7	BACnet Test Server ▾	0.00	AI 7	Analog Input 7	
8	Analog Input ▾	8	BACnet Test Server ▾	0.00	AI 8	Analog Input 8	
9	Analog Input ▾	9	BACnet Test Server ▾	0.00	AI 9	Analog Input 9	
10	Analog Input ▾	10	BACnet Test Server ▾	0.00	AI 10	Analog Input 10	
11	Analog Input ▾	11	BACnet Test Server ▾	0.00	AI 11	Analog Input 11	
12	Analog Input ▾	12	BACnet Test Server ▾	0.00	AI 12	Analog Input 12	
13	Analog Input ▾	13	BACnet Test Server ▾	0.00	AI 13	Analog Input 13	
14	Analog Input ▾	14	BACnet Test Server ▾	0.00	AI 14	Analog Input 14	
15	Analog Input ▾	15	BACnet Test Server ▾	0.00	AI 15	Analog Input 15	

7.6 Clearing Configuration

Read and write maps imported from a CSV file will be added to the list of maps already in place. If you wish to reload the list, you must first clear it. Clear the BACnet client maps by going to the File Manager page, then selecting "Clear BACnet Client Maps" from the action list and clicking Execute.

File Manager	Network	Resources	User
Free space: 1.37 MB			
File Directory: P3TR-bip-maps.csv		Filtered by: *.csv Filter View Select	
Selected File: P3TR-bip-maps.csv		Action: Clear BACnet Client Maps Execute	
Boot configuration: BootConfig.xml		Confirm Restart	
Upload File Upload Browse... No file selected.			

If you forget to clear the maps before re-importing them, you will get an error notice something like this:



The following error(s) occurred:

```
Line 2 Col 18: Local object already mapped
Line 3 Col 18: Local object already mapped
Line 4 Col 18: Local object already mapped
Line 5 Col 18: Local object already mapped
Line 6 Col 18: Local object already mapped
Line 7 Col 18: Local object already mapped
```

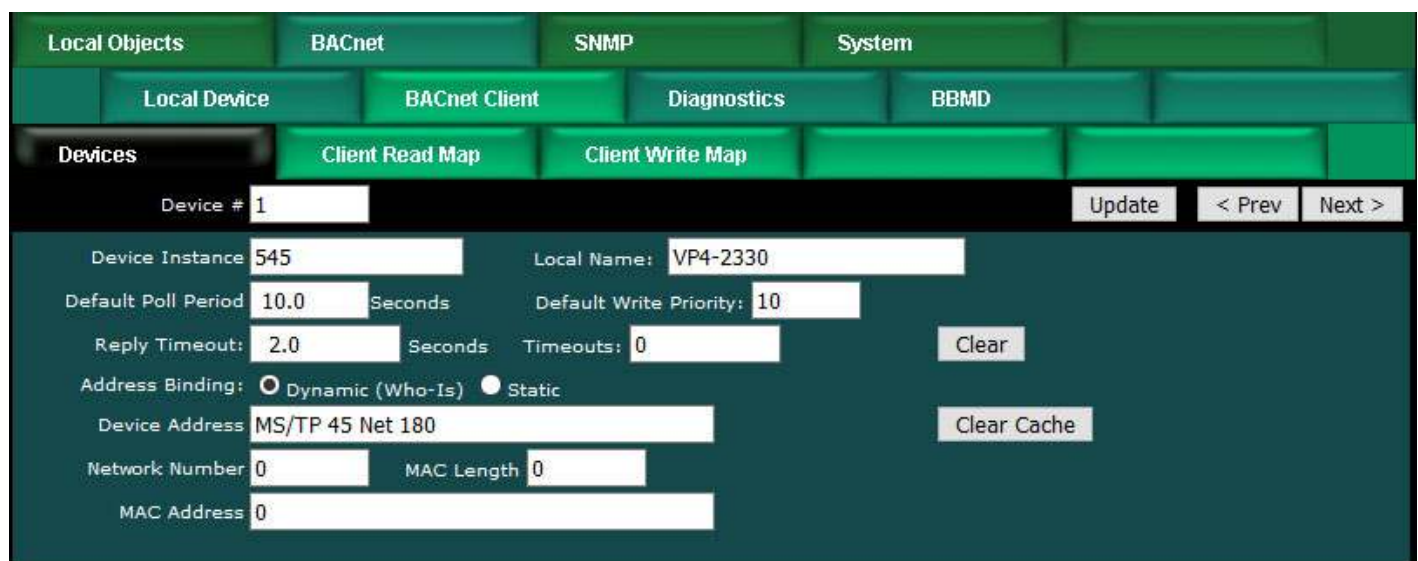
The error is most likely the result of an incorrect entry in one or more fields of a form. Click your browser's "back" button to go back to the page you were at, make corrections, and try again.

7.7 Understanding BACnet Client Timeout Settings

The Babel Buster gateway includes a BACnet client that can be configured to read and write objects in other BACnet devices. Each point to be read or written is defined by a client read map or client write map. These maps reference a device, and each device has an entry on the Devices page. For each device, the BACnet Device Instance, a name, default poll period, and timeout are provided by the user.

The Reply Timeout is the amount of time the client will wait for a response before calling it a timeout if no response is received. The client will then move on to the next read or write map. The client will eventually come back around to the same point and try again. If the client times out a second time, then the mapped object's reliability code will be set to the nonzero value indicating timeout, no response.

If repeated timeouts are observed, one should confirm that the device in question is operating. If so, then set a longer timeout period as needed.



The screenshot shows the configuration interface for the BACnet Client. The top navigation bar includes tabs for Local Objects, BACnet, SNMP, and System. The BACnet tab is active, showing sub-tabs for Local Device, BACnet Client, Diagnostics, and BBMD. The BACnet Client sub-tab is selected, displaying the configuration for Device #1. The interface includes fields for Device Instance (545), Local Name (VP4-2330), Default Poll Period (10.0 Seconds), Default Write Priority (10), Reply Timeout (2.0 Seconds), Timeouts (0), Address Binding (Dynamic (Who-Is) selected), Device Address (MS/TP 45 Net 180), Network Number (0), MAC Length (0), and MAC Address (0). There are buttons for Update, < Prev, Next >, Clear, and Clear Cache.

Much of the time, especially with BACnet IP, the client device timeout is the only timeout one needs to be pay much attention to. However, it is important to understand what is going on inside BACnet behind the scenes, especially if the client is making requests to an MS/TP device on the other side of a router.

The BACnet Device has settings that apply to all requests made by this device, and these are found on the BACnet settings page (illustrated below). Of particular concern with respect to timeouts are the APDU Timeout and APDU Retries.

Any time a request is made by this BACnet device, the request initiates a Transaction State Machine (TSM). The Invoke ID you will see if you use Wireshark to look at network traffic identifies this TSM. This invoke ID is used to associate a reply with a request. If the TSM does not see a reply within the APDU Timeout (given in milliseconds), then the TSM will automatically retransmit the request and wait another APDU Timeout period. This retransmission will be repeated up to the retry count. If the retry count is 3 as illustrated below (with a timeout of 3000 milliseconds), and no reply is ever received, this means the request will have been transmitted a total of 4 times (over 12 seconds).

The APDU Timeout will default to 3000 (milliseconds) and APDU Retries will default to 3 as recommended by BACnet protocol. However, whether these numbers make sense for your application is left for you to determine.

It makes sense to have the BACnet client device timeout set to 2 seconds as illustrated above so that a timeout doesn't hang up the client for too long. However, if the default values for the BACnet Device are left as illustrated below, then here is what is going to happen when the target device does not reply: The client will send its initial request. Then 2 seconds later, it decides there is no response and moves on to the next point and sends the next request. Meanwhile the TSM has waited 3 seconds, then retransmitted the original request. Over a period of 12 seconds, the TSM will be sending the same original first request 4 times. As you can see, the client has not really waited for the final result in this instance. Furthermore, the client is kicking off more TSM's faster than they can complete their retry sequence. If the target device is a slow MS/TP device on the other side of a router, it is highly likely that you will flood the router with more requests than it can handle, and you will start to get "reject - router busy" replies from the router (which will be indicated simply as timeout on the client end).

This snowball effect and request log jam will often clear itself when the BACnet client is only polling BACnet IP devices. But the snowball effect can have very adverse effects on a slow MS/TP network on the other side of a router.

When choosing a timeout value for the devices listed on the Devices page in the BACnet client, be sure to also examine the APDU Timeout and Retries on the BACnet settings page. If the default values illustrated below are left as is, then the most suitable timeout value for the client device above would be 12 seconds, not 2 seconds.

BACnet Settings	Router Settings	Configured Routes	Discovered Routes	Slave Proxy
BACnet Device Settings: <div>Local Network Settings</div> <div> <div>Device Instance: 208078</div> <div>Port (default 0xBAC0 = 47808): 47808</div> <div>Device Object Name: Babel Buster BB3-7302-V2C</div> <div>Device Description:</div> <div>Device Location: St. Paul, Minnesota</div> <div>APDU Timeout: 3000</div> <div>APDU Segment Timeout: 5000</div> <div>Local Command Priority: 10</div> <div>MS/TP Baud Rate: 38400</div> <div>MS/TP Max Info Frames: 10</div> <div>APDU Retries: 3</div> <div>Database Revision: 14</div> <div>Backlog Limit: 0</div> <div>MS/TP MAC address: 78</div> <div>MS/TP Max Masters: 127</div> <div>Save</div> </div>				

Another setting one should pay attention to especially when talking to MS/TP devices on the other side of a router is the "Backlog Limit". The BACnet client will not necessarily wait for the reply from device A before sending a request to device B. If the client is polling 20 different devices, it is quite plausible that the client will send 20 requests faster than the first reply can come back. Thus it is quite easy for a BACnet IP client to overrun an MS/TP router by pumping out requests faster than the router can forward them to MS/TP. Therefore, one should use the Backlog Limit to throttle the client. If the limit is set to 4 (a reasonable number for MS/TP), this means the client will send no more than 4 requests before pausing and waiting for replies to those requests.

The other aspect of how Backlog Limit may affect required timeout setting is that when there is a large backlog of replies to process, the remote device may have responded promptly and within the client timeout setting, but by the time the client gets through the backlog of replies, a short client timeout may have expired. Therefore, timeout is not just a matter of how fast the other device responds, but also a matter of how busy you are keeping the client.

Finally, in the case of an MS/TP router, the MS/TP settings include Max Info Frames. This is the number of requests that the MS/TP device may make before it is required to pass the token. Setting this to some high number will greatly increase throughput in forwarding requests from the IP side to the MS/TP side, and will have a significant impact on timeout settings required in any IP client that is routing its requests to MS/TP devices through this router.



8. Configuring Gateway as a BACnet Server

8.1 Server Configuration

The BB3-7302/MX-73 contains a set of BACnet objects whose only purpose is to store copies of data obtained from other devices. This copy of data may then be queried by different devices.

The only configuration needed to use the BB3-7302/MX-73 as a BACnet server is to set the Device instance on the BACnet page. The device should also be given an object name that will be unique on the entire network. Configuring the gateway as a BACnet Device is described in more detail in Section 4.

Babel Buster[®] 3
BACNET
ROUTER & GATEWAY
MODEL BB3-7302-V2C

CONTROL SOLUTIONS MINNESOTA

Local Objects | BACnet | SNMP | System

Local Device | BACnet Client | Diagnostics | BBMD

BACnet Settings | Router Settings | Configured Routes | Discovered Routes | Slave Proxy

BACnet Device Settings: Local Network Settings

Device Instance: 208078

Port (default 0xBAC0 = 47808): 47808

Device Object Name: Babel Buster BB3-7302-V2C

Device Description:

Device Location: St. Paul, Minnesota

APDU Timeout: 3000

APDU Retries: 3

APDU Segment Timeout: 5000

Database Revision: 5

Local Command Priority: 10

Backlog Limit: 0

MS/TP Baud Rate: 38400

MS/TP MAC address: 78

MS/TP Max Info Frames: 10

MS/TP Max Masters: 127

Allow fault self-reset without Ack. ☒

Disable self-restart upon communications loss ☒

Disable Segmentation. ☐

Slave Proxy Mode: ☐ None ☐ Manual ☐ Auto

Save

8.2 Accessing Local Objects

The collection of local objects includes Analog, Binary, and Multi-State types of objects, and includes Input, (commandable) Output, and (writeable) Value types of each of those objects. The BB3-7302/MX-73 also contains a Device object which is configured in the above screen.

Data may be placed in the local objects by other devices writing to the BB3-7302/MX-73, or by the BB3-7302/MX-73 querying other devices. When the BB3-7302/MX-73 is configured to query other devices, these operations are defined by "read maps" and "write maps" associated with the respective client function (BACnet client or SNMP client).

The following pages illustrate the Analog Input object pages and the Binary Output object pages. The remaining object pages found in the BB3-7302/MX-73 are virtually identical, and are not replicated here. (See also Configuring Local Objects, Section 6.)

Each object page initially comes up as a table of object data. Click on the object number in the left-hand column to expand the view of that object and access the windows that let you locally force values, assign units or names, etc.

Local Objects

BACnet

SNMP

System

Analog

Binary

Multi-State

Actions

Input Objects

Output Objects

Value Objects

Analog Input Objects

Showing objects from1

Refresh

< Prev

Next >

Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Units
<u>1</u>	Analog Input 1 Description of AI 1	N	0.00	0	0,0,0,0	no_units
<u>2</u>	Analog Input 2 Description of AI 2	N	0.00	0	0,0,0,0	no_units
<u>3</u>	Analog Input 3 Description of AI 3	N	15.000000	0	0,0,0,0	no_units
<u>4</u>	Analog Input 4 Description of AI 4	N	0.00	0	0,0,0,0	no_units
<u>5</u>	Analog Input 5 Description of AI 5	N	0.00	0	0,0,0,0	no_units
<u>6</u>	Analog Input 6 Description of AI 6	N	0.00	0	0,0,0,0	no_units
<u>7</u>	Analog Input 7 Description of AI 7	N	0.00	0	0,0,0,0	no_units

The object name, units, value, and status are shown for a list of objects starting with the number entered at the top of the page. Click Prev/Next to scroll through the list. Click on the object number in the first column to change name, units, COV, and out-of-service state.

The source of data for an Analog Input object will typically be reading from some other BACnet or SNMP device. Click on the object number in the first column for more detail including the link to any client map providing data to this object.

Out of Service means polling for data will stop. While out of service, the present value may be written by the BACnet client. Data may be forced via this web page at any time, but will be overwritten by the next mapped client update unless the object is out of service.

Reliability codes indicate error conditions if non-zero, and may be any of the following:

BACnet IP client, device timeout (82)
 BACnet IP client, error returned by server (83)
 SNMP client, no response from agent (91)
 SNMP client, agent returned 'no such name' error (92)
 SNMP client, unable to parse data returned (93)
 SNMP client, reply did not match request (94)
 SNMP client, invalid table walk configuration (95)
 SNMP client, unable to parse PDU returned (96)
 SNMP client, table walk came up short on data (97)
 SNMP client, agent returned error other than 'no such name' (98)

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

A = in alarm
 B = fault
 C = overridden
 D = out of service

Click on an Object number in the first column of maps to get the expanded view of

that object as follows:

The screenshot shows a web interface for configuring BACnet objects. At the top, there are tabs for 'Local Objects', 'BACnet', 'SNMP', and 'System'. Under 'BACnet', there are sub-tabs for 'Analog', 'Binary', 'Multi-State', and 'Actions'. The 'Analog' tab is selected, and within it, 'Input Objects' is chosen. The main area displays the configuration for 'Analog Input # 3'. Fields include: 'Reliability: 0', 'Status: 0,0,0,0', 'Device Link: ---', 'Out of Service' (checkbox), 'Deconfigure' (checkbox), 'Object name: Analog Input 3', 'Description: Description of AI 3', 'COV increment: 0.00', 'Units: no_units', 'Force' (checkbox), and 'Present Value: 15.00000'. Navigation buttons 'Update', '< Prev', and 'Next >' are at the top right.

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page, or Update to accept changes.

The object name and description may be changed here. BACnet units may be selected. Initial COV increment may be entered. When any of these are changed, be sure to save the updated configuration by clicking Save on the Config File page under System Setup.

The object may be set Out of Service by checking that box and clicking Update. The present value may be changed by entering a value, checking Force, and clicking Update.

The source of data for an Analog Input object will typically be reading from some other BACnet or SNMP device via the map indicated by the Device Link. The mapped device will be polled at the rate specified by the Read Map.

Out of Service means polling of the mapped remote device will stop. While out of service, the present value may be written by the BACnet client. Data may be forced via this web page at any time, but will be overwritten by the next mapped client update unless the object is out of service.

Reliability codes indicate errors as itemized on the tabular object list.

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

- A = in alarm
- B = fault
- C = overridden
- D = out of service

Device link will indicate BIP or SNMP, followed by R for read or W for write, and a number which is the map number in the table of read or write maps for mapping to this BACnet object. The designation R means read from a remote device, and W means write to a remote device.

Check 'Deconfigure' and click Update to erase configuration for this object.

Local Objects		BACnet		SNMP		System			
Analog		Binary		Multi-State		Actions			
Input Objects		Output Objects		Value Objects					
Binary Output Objects									
				Showing objects from		1		Update	
		< Prev						Next >	
Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Text			
1	Relay Output 1 Remote relay control	N	Active	0	0,0,0,0	Relay closed			

The object name, value, and status are shown for a list of objects starting with the number entered at the top of the page. Click Prev/Next to scroll through the list. Click on the object number in the first column to change name or out-of-service state.

The destination of data for a Binary Output object will typically be some other BACnet or SNMP device. Click on the object number in the first column for more detail including the link to any client map receiving data from this object.

The Binary Output object is commandable, meaning the BACnet client must write both a value and a priority level for that value. The highest level value will be the one written to the remote device (if mapped). If all values are relinquished, the relinquish default value will be written to the remote device.

Out of service means the mapped remote device will not be written to. Values written by the BACnet client will be retained, but only applied when this object is placed back in service. At that time, the highest priority value will be written to the remote device.

Reliability codes indicate error conditions if non-zero, and may be any of the following:

BACnet IP client, device timeout (82)
 BACnet IP client, error returned by server (83)
 SNMP client, no response from agent (91)
 SNMP client, agent returned 'no such name' error (92)
 SNMP client, unable to parse data returned (93)
 SNMP client, reply did not match request (94)
 SNMP client, invalid table walk configuration (95)
 SNMP client, unable to parse PDU returned (96)
 SNMP client, table walk came up short on data (97)
 SNMP client, agent returned error other than 'no such name' (98)

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

A = in alarm
 B = fault
 C = overridden
 D = out of service

Click on an Object number in the first column of maps to get the expanded view of that object as follows:

The screenshot shows the configuration page for a Binary Output object. At the top, there are tabs for 'Local Objects', 'BACnet', 'SNMP', and 'System'. Under 'BACnet', there are sub-tabs for 'Analog', 'Binary', 'Multi-State', and 'Actions'. The 'Binary' sub-tab is selected, and under it, 'Output Objects' is chosen. The 'Binary Output #' is set to 1. Buttons for 'Update', '< Prev', and 'Next >' are visible. The object name is 'Relay Output 1', description is 'Remote relay control', active text is 'Relay closed', and inactive text is 'Relay open'. The 'Relinquish Default' is set to 'Inactive'. A dropdown menu for 'Present Value' is open, showing a list of values from 1 to 16, with '4> Active' selected. Below the configuration fields is a 'Quick Help' section with three paragraphs of text explaining the object's behavior and how to use the interface.

Local Objects BACnet SNMP System

Analog Binary Multi-State Actions

Input Objects **Output Objects** Value Objects

Binary Output # 1 Update < Prev Next >

Reliability: 0 Status: 0,0,0,0 Device Link: --- Out of Service: ☐ Deconfigure: ☐

Object name: Relay Output 1 Force ☐ Present Value: Active 4> Active

Description: Remote relay control

Active Text: Relay closed Inactive Text: Relay open

Relinquish Default: Inactive

Quick Help

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page, or Update to accept changes.

The object name and description may be changed here. State text may be entered. When any of these are changed, be sure to save the updated configuration by clicking Save on the Config File page under System Setup.

The destination of data for a Binary Output object will be writing the remote BACnet or SNMP device via the map indicated by the Device Link. The remote device will be updated upon change of source data and/or periodically as defined by the Write Map.

The object name, units, value, and status are shown for the object number entered at the top of the page. Click Prev/Next to scroll through the list. Click Refresh to update the page, or Update to accept changes.

The object name and description may be changed here. State text may be entered. When any of these are changed, be sure to save the updated configuration by clicking Save on the Config File page under System Setup.

The destination of data for a Binary Output object will be writing the remote BACnet or SNMP device via the map indicated by the Device Link. The remote device will be updated upon change of source data and/or periodically as defined by the Write Map.

The Binary Output object is commandable, meaning the BACnet client must write both a value and a priority level for that value. The highest level value will be the one written to the remote device (if one is mapped). If all values are relinquished, the relinquish default value will be written to the remote device.

To set an output object manually from this page, check the Force box, enter a value in the Present Value window, and select a priority level to assign to your forced value. Then click Update. To return a given priority level to NULL, simply type the word NULL in the Present Value window, check Force, and click Update.

Out of service means the mapped remote device will not be written to. Values written by the BACnet client will be retained, but only applied when this object is placed back in service. At that time, the highest priority value will be written to the remote device.

Reliability codes indicate errors as itemized on the tabular object list.

Status flags A,B,C,D indicate the following, 0 meaning not true, 1 meaning true:

A = in alarm

B = fault

C = overridden

D = out of service

Device link will indicate BIP or SNMP, followed by R for read or W for write, and a number which is the map number in the table of read or write maps for mapping to this BACnet object. The designation R means read from a remote device, and W means write to a remote device.

Check 'Deconfigure' and click Update to erase configuration for this object.



9. Configuring BBMD

BBMD stands for BACnet Broadcast Management Device. Messages such as "Who-Is" and "I-Am" are broadcast. Most NAT routers, however, do not pass broadcast messages along. The BBMD solves this problem by explicitly directing broadcast messages to a specific IP address.

The image is a screenshot of the 'BBMD Settings' page in the Babel Buster 3 software. The page has a dark green background with a grid of buttons at the top. The buttons are labeled 'Local Objects', 'BACnet', 'Modbus', 'System', 'Local Device', 'BACnet Client', 'Diagnostics', 'BBMD', 'BBMD Settings', and 'Edit BDT'. The 'BBMD Settings' button is highlighted. Below the buttons, there is a 'Refresh' button. The main content area is divided into three sections. The first section is titled 'Broadcast Distribution Table (BDT)' and contains a table with three columns: 'Broadcast Address:Port', 'Broadcast Mask', and an empty column. The second section is titled 'Foreign Devices Registered Locally' and contains a table with three columns: 'Foreign Devices Registered Locally', 'Time to Live', and an empty column. The third section is titled 'Local Device's Registration as a Foreign Device at Remote Location' and contains a form with the following fields: 'Enable BBMD' (a checkbox), 'BBMD Time To Live (seconds)' (a text box with '0' and a note '(Zero disables foreign registration)'), 'BBMD IP Address, Port' (a text box with '0.0.0.0' and a port box with '0'), and a 'Save' button. The status 'BBMD is not registered.' is displayed at the bottom right of the form.

The BBMD Settings page appears as shown above when no part of BBMD support is enabled, as is the case when shipped. Do not enable BBMD if you are not aware of needing it and/or do not understand how BBMD works. The three elements of BBMD support are discussed in the following sections, and their use is often mutually exclusive, meaning you will often need only one of the three elements.

9.1 Registering as a Foreign Device

If you have a remote BB3-7302/MX-73 that needs to connect via router, including NAT router, to a local network, use Foreign Device Registration. There will typically be a master device, such as operator station or other front end, that includes BBMD. The IP address of this device is the one that should be given as the BBMD address for foreign device registration.

The screenshot shows a web-based configuration interface for BBMD. The top navigation bar includes tabs for 'Local Objects', 'BACnet', 'Modbus', and 'System'. Under the 'Local Objects' tab, there are sub-tabs for 'Local Device', 'BACnet Client', 'Diagnostics', and 'BBMD'. The 'BBMD Settings' section is active, showing a 'Refresh' button and a table for the 'Broadcast Distribution Table (BDT)'. Below this is another 'Refresh' button and a table for 'Foreign Devices Registered Locally'. The bottom section, 'Local Device's Registration as a Foreign Device at Remote Location', contains a checkbox for 'Enable BBMD' (checked), a text input for 'BBMD Time To Live (seconds)' set to 900, a 'Save' button, and a text input for 'BBMD IP Address, Port' set to 173.22.32.91 and 47808. A status message 'BBMD is registered.' is displayed.

Broadcast Distribution Table (BDT)	Broadcast Address:Port	Broadcast Mask
	---	---

Foreign Devices Registered Locally	Time to Live
---	---

Local Device's Registration as a Foreign Device at Remote Location

Enable BBMD ☒

BBMD Time To Live (seconds) (Zero disables foreign registration)

BBMD IP Address, Port BBMD is registered.

To enable BBMD processing, check the "Enable BBMD" box. This applies to foreign device registration. The broadcast distribution table functions regardless of whether foreign device registration is enabled.

If the BB3-7302/MX-73 should register as a foreign device with another BBMD, then the port, time-to-live, and IP address of the remote BBMD must be given. The local BBMD will attempt to register with the remote BBMD whose address is given.

Disable this device's attempts to register elsewhere, but allow other devices to register here, by setting time to live to zero with BBMD enabled.

9.2 Allowing Other Devices to Register Locally

The BB3-7302/MX-73 can be the BBMD that other devices register with. The screen shot below shows that three other devices have registered with this BBMD, and broadcast messages will now be sent explicitly to these locations. In this case, there are NAT routers between this local device and the three remote devices. While they are all on physically separate local networks, they will appear as a single BACnet network even if the local networks are miles apart. The local BACnet client will be able to communicate with these remote BACnet devices as a result of the foreign registration.

Note that foreign registration only provides communication with a single remote device. If communicating with an entire remote network of BACnet devices is the intent, then full BACnet routing is required and the foreign device should be another BACnet router.

The screenshot displays the BBMD configuration interface. At the top, there are tabs for 'Local Objects', 'BACnet', 'Modbus', and 'System'. Under the 'BACnet' tab, there are sub-tabs for 'Local Device', 'BACnet Client', 'Diagnostics', and 'BBMD'. The 'BBMD' sub-tab is selected, showing 'BBMD Settings' and an 'Edit BDT' button. A 'Refresh' button is located to the right of the settings. Below this is the 'Broadcast Distribution Table (BDT)' with columns for 'Broadcast Address:Port' and 'Broadcast Mask'. The table is currently empty. Another 'Refresh' button is below the table. Below the table is a section for 'Foreign Devices Registered Locally' with columns for 'Foreign Devices Registered Locally' and 'Time to Live'. This table contains three entries, all with a 'Time to Live' of 630. At the bottom is the 'Local Device's Registration as a Foreign Device at Remote Location' section. It includes a checkbox for 'Enable BBMD' which is checked. Below it is a text input for 'BBMD Time To Live (seconds)' set to 0, with a note '(Zero disables foreign registration)'. To the right is a 'Save' button. Below that is a text input for 'BBMD IP Address, Port' set to '0.0.0.0' and '0', with a status message 'BBMD is not registered.'

Broadcast Distribution Table (BDT)	Broadcast Address:Port	Broadcast Mask
	---	---

Foreign Devices Registered Locally	Time to Live
173.22.32.87:47808	630
173.22.32.90:47808	630
173.22.32.91:47808	630

Local Device's Registration as a Foreign Device at Remote Location

Enable BBMD ☒

BBMD Time To Live (seconds) (Zero disables foreign registration)

BBMD IP Address, Port BBMD is not registered.

To allow foreign devices to register with this device, but not have this device register elsewhere, check Enable BBMD, but enter zero for BBMD Time To Live. This enables BBMD but disables this device's attempt to register somewhere else.

9.3 Broadcast Distribution Table

A Broadcast Distribution Table (BDT) defines a list of IP addresses that the BBMD should send broadcast messages to. It is important to note that a BBMD only forwards broadcast messages. The router is responsible for forwarding all other messages.

Broadcast distribution will result in device discovery, but you will not be able to read/write properties in the remote device without full routing. Foreign device registration via a router does result in being able to fully communicate with the foreign device from the local network.

Local Objects	BACnet	Modbus	System
Local Device	BACnet Client	Diagnostics	BBMD
BBMD Settings	Edit BDT		

Update

	Broadcast Address : Port		Broadcast Mask	
	192.168.1.126	47808	24	FFFFFF00
	173.22.32.87	47808	32	FFFFFFF
	173.22.32.90	47808	32	FFFFFFF
	173.22.32.91	47808	32	FFFFFFF
	0.0.0.0	0	0	00000000

The Edit BDT page allows viewing of the broadcast distribution table that has been provided to the local device by an external network management tool capable of sending the BDT initialize. The BDT may also be edited on this page. Regardless of how the table is filled, it will be saved in the configuration file when saved, and reloaded upon restart.

Once the table has been initialized, it will appear on the BBMD Settings page as illustrated below.

Local Objects	BACnet	Modbus	System
Local Device	BACnet Client	Diagnostics	BBMD
BBMD Settings	Edit BDT		

Refresh

Broadcast Distribution Table (BDT)	Broadcast Address:Port	Broadcast Mask
	192.168.1.126:47808	FFFFFF00
	173.11.32.87:47808	FFFFFFF
	173.11.32.90:47808	FFFFFFF
	173.11.32.91:47808	FFFFFFF

Refresh

	Foreign Devices Registered Locally	Time to Live
	---	---

Local Device's Registration as a Foreign Device at Remote Location

Enable BBMD ☒

BBMD Time To Live (seconds) (Zero disables foreign registration)

Save

BBMD IP Address, Port BBMD is not registered.



10. Configuring Gateway as an SNMP Server

10.1 Creating Local SNMP MIB

The Babel Buster BB3-7302/MX-73 starts out with no variables in its MIB. There are two branches in the BB3 MIB that are accessible to BACnet, although only the Integer branch is guaranteed to be universally accessible to all other SNMP devices. It is up to you to select which branch of the MIB to place each local object in. You do not need to place all local objects in the MIB, only those that you want externally accessible via SNMP. You must also place local objects in the MIB in order to generate SNMP traps related to those local objects.

Map #	Local SNMP OID	Local Object	Scale Factor	Local Value	Local Name
1	1.3.6.1.4.1.3815.1.6.1.1.1.2.1	None	x1	0	---

To add a local object to a MIB branch, simply enter the local object number at the next available OID which will always automatically be at the bottom of the list. When placing objects in the Integer branch, you also have the option of applying a scale factor. Scaled integer is the most universally recognized means of transmitting non-integer data.

Local Objects BACnet **SNMP** System

Local MIB Client Setup Client Data Trap Sender Trap Receiver

Integer 32-bit **Float 32-Bit**

Showing 1 to 1 of 1 Update < Prev Next >

Map #	Local SNMP OID	Local Object	Scale Factor	Local Value	Local Name
1	1.3.6.1.4.1.3815.1.6.1.1.1.2.1	AI 1	x1	0	---

Auto-fill 0 variables starting at index 1 with local object AI 1 Auto-Fill Confirm Clear

Reload SNMP Map # 1 Remove Insert Before

Quick Help

Map number simply tells you where you're at on the list of the...
list. To advance directly to a specific map, enter the desired n...
"showing" box, then click Update.

This page enables SNMP Get/Set to objects indicated on the s...
select which local objects are mapped to these OID's.

Internal data is multiplied by the scale factor when read by yo...
by the scale factor before being stored internally. If the object...
scaling). However, if the object is a floating point value, then...
is the one universally recognized data type in SNMP. While floa...
Therefore, the oldest and best known recommendation for se...
send them as ASCII strings (supported by the Char String bra...

...ant's OID maps. Click "next" and "prev" to scroll through the...
...the available local OID's are assigned automatically. You may...
...manager (client). Data written by your SNMP client is divided...
...n integer value, then scale factor will most often be x1 (no...
...becomes more important. This is significant because INTEGER...
...presentations do exist, they are not universally accepted.
...ers via SNMP is to scale them and send them as integer, or...

Local Objects BACnet **SNMP** System

Analog Binary Multi-State Actions

Input Objects **Output Objects** Value Objects

Analog Input Objects Showing objects from 1 Refresh < Prev Next >

Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Units
1	Analog Input 1	N	5.190000	0	0,0,0,0	no_units
2	Analog Input 2	N	0.00	0	0,0,0,0	no_units

In this example, we have selected a scale factor of x100. That means that our local value of 5.19 will be transmitted (in a Get response) as integer 519 and it is up to the recipient to know that this is scaled x100.

Integer 32-bit **Float 32-Bit**

Showing 1 to 2 of 2 Update < Prev Next >

Map #	Local SNMP OID	Local Object	Scale Factor	Local Value	Local Name
1	1.3.6.1.4.1.3815.1.6.1.1.1.2.1	AI 1	x100	519	Analog Input 1
2	1.3.6.1.4.1.3815.1.6.1.1.1.2.2	None	x1	0	---

Auto-fill 0 variables starting at index 1 with local object AI 1 Auto-Fill Confirm Clear

Reload SNMP Map # 1 Remove Insert Before

After adding new members to the MIB, it is necessary to click Reload SNMP before

they will become accessible to an external SNMP manager's Get request.

Map #	Local SNMP OID	Local Object	Scale Factor	Local Value	Local Name
1	1.3.6.1.4.1.3815.1.6.1.1.1.1.2.1	AI 1	x100	519	Analog Input 1
2	1.3.6.1.4.1.3815.1.6.1.1.1.1.2.2	None	x1	0	---

Auto-fill 0 variables starting at index 1 with local object AI 1

Reload SNMP 1 Int32 vars loaded

Map # 1

Local objects added to the Float 32-Bit MIB branch will be provided in IEEE 754 format per RFC 6340.

The "Auto-fill" at the bottom of each MIB page can be used to very quickly create a long list of sequential MIB table entries. Enter the number of variables to fill, the starting MIB index, and the starting local object number. Then click the Auto-Fill button.

Map #	Local SNMP OID	Local Object	Scale Factor	Local Value	Local Name
1	1.3.6.1.4.1.3815.1.6.1.1.1.1.2.1	None	x1	0	---

Auto-fill 10 variables starting at index 1 with local object AI 1

Auto-Fill

Reload SNMP

Map # 1

The example above will create the list illustrated below, assuming the first ten local objects are all 16-bit integer objects.

Integer 32-bit

Float 32-Bit

Showing 1 to 11 of 11

Update < Prev Next >

Map #	Local SNMP OID	Local Object	Scale Factor	Local Value	Local Name
1	1.3.6.1.4.1.3815.1.6.1.1.1.2.1	AI 1	x1	0	Analog Input 1
2	1.3.6.1.4.1.3815.1.6.1.1.1.2.2	AI 2	x1	0	Analog Input 2
3	1.3.6.1.4.1.3815.1.6.1.1.1.2.3	AI 3	x1	0	Analog Input 3
4	1.3.6.1.4.1.3815.1.6.1.1.1.2.4	AI 4	x1	0	Analog Input 4
5	1.3.6.1.4.1.3815.1.6.1.1.1.2.5	AI 5	x1	0	Analog Input 5
6	1.3.6.1.4.1.3815.1.6.1.1.1.2.6	AI 6	x1	0	Analog Input 6
7	1.3.6.1.4.1.3815.1.6.1.1.1.2.7	AI 7	x1	0	Analog Input 7
8	1.3.6.1.4.1.3815.1.6.1.1.1.2.8	AI 8	x1	0	Analog Input 8
9	1.3.6.1.4.1.3815.1.6.1.1.1.2.9	AI 9	x1	0	Analog Input 9
10	1.3.6.1.4.1.3815.1.6.1.1.1.2.10	AI 10	x1	0	Analog Input 10
11	1.3.6.1.4.1.3815.1.6.1.1.1.2.11	None	x1	0	---

Auto-fill 10 variables starting at index 1 with local object AI 1

Auto-Fill

Confirm

Clear

Reload SNMP

Map # 1

Remove

Insert Before

The Auto-Fill simply sets up the MIB mapping table. Before the variables can be accessed by SNMP, you need to click Reload SNMP. The Remove and Insert Before buttons also just set up the mapping table, and you need to click Reload SNMP to make the new assignments accessible.

To clear the mapping table displayed, click the Confirm box and then click the Clear button. This clears the mapping table, but as with Auto-Fill, no changes (from the perspective of SNMP as viewed from the outside) take place until you click Reload SNMP. The Clear button is mainly useful if you change your mind about the auto-fill you just did.

Due to SNMP's memory allocation scheme, you can always increase the size of the MIB by simply clicking Reload SNMP; however, to reduce the size of the MIB, you need to save your configuration on the File Manager page, and then restart the gateway. If you reduce the number of entries in the mapping table, the original MIB variables at the end of the original list will simply return zero or empty strings until the gateway is restarted.

10.2 Supported Data Formats, RFC 6340

SNMP does not have a universally accepted representation for floating point. The one universally known data type is INTEGER. A commonly recommended means of transmitting floating point data is either as a scaled integer or as an ASCII character string. There is an RFC 6340 for representation of floating point based on IEEE 754 encoding. The "Float 32-bit" and "Float 64-bit" data types in the Babel Buster refer to RFC 6340 encoding.

Specifically, the data types found in the Babel Buster MIB are encoded with ASN types as follows:

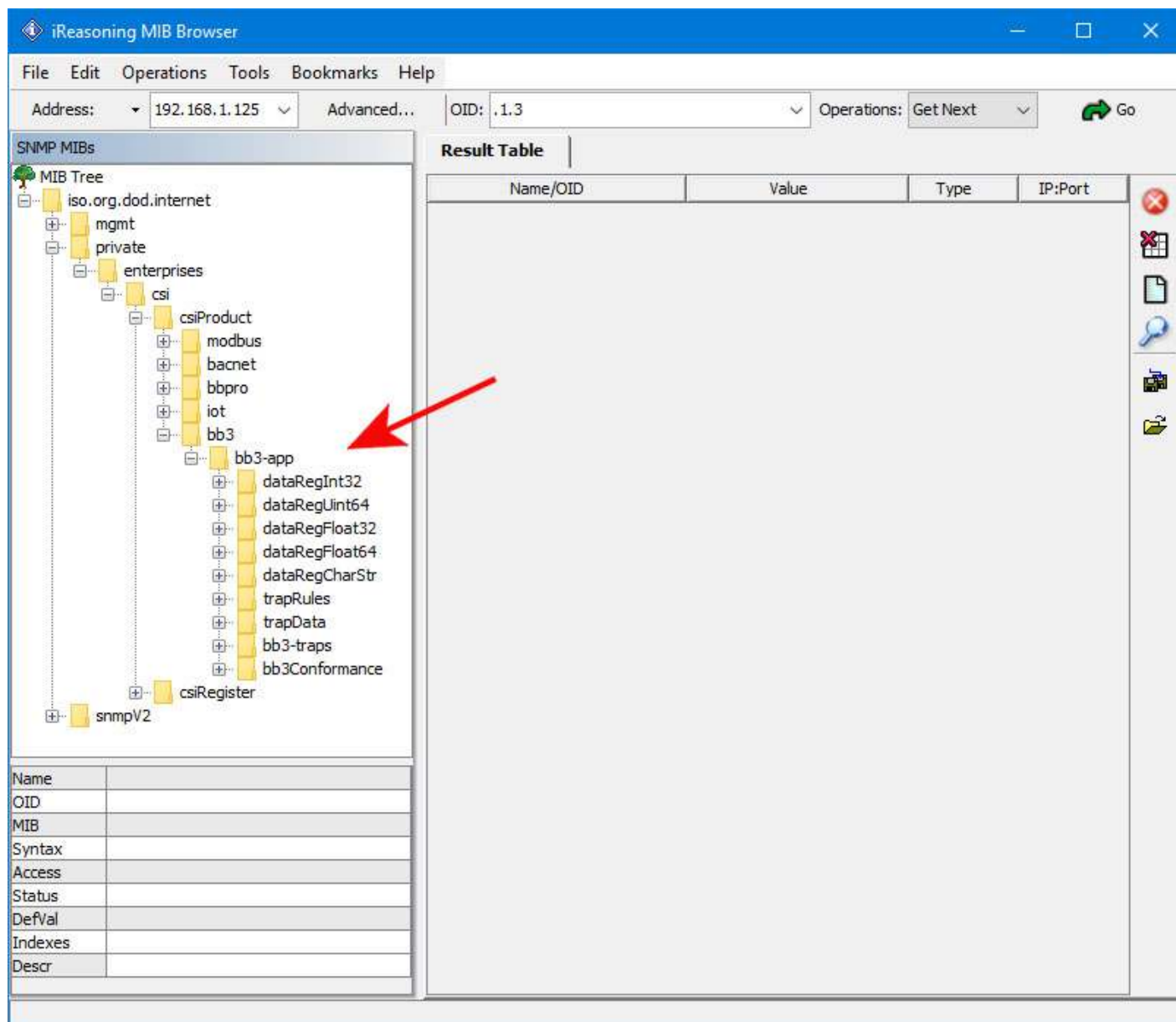
Integer 32-Bit	INTEGER	ASN_INTEGER
Float 32-bit	OCTET STRING	ASN_OCTET_STR (length 4)

Note: The Babel Buster 3 MIB defines additional branches in the MIB for additional data types; however, these are not used by the BB3-7302/MX-73.

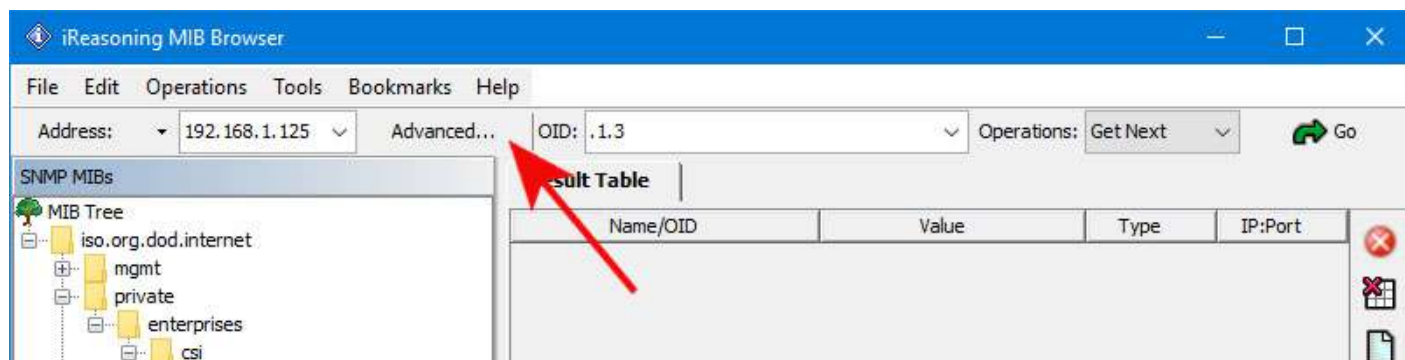
10.3 Testing the SNMP Agent

A variety of tools are available for browsing an SNMP MIB and receiving SNMP Traps. The tool used in the following examples is the iReasoning MIB Browser. Refer to the Tools section under Support at csimn.com for more information about SNMP tools.

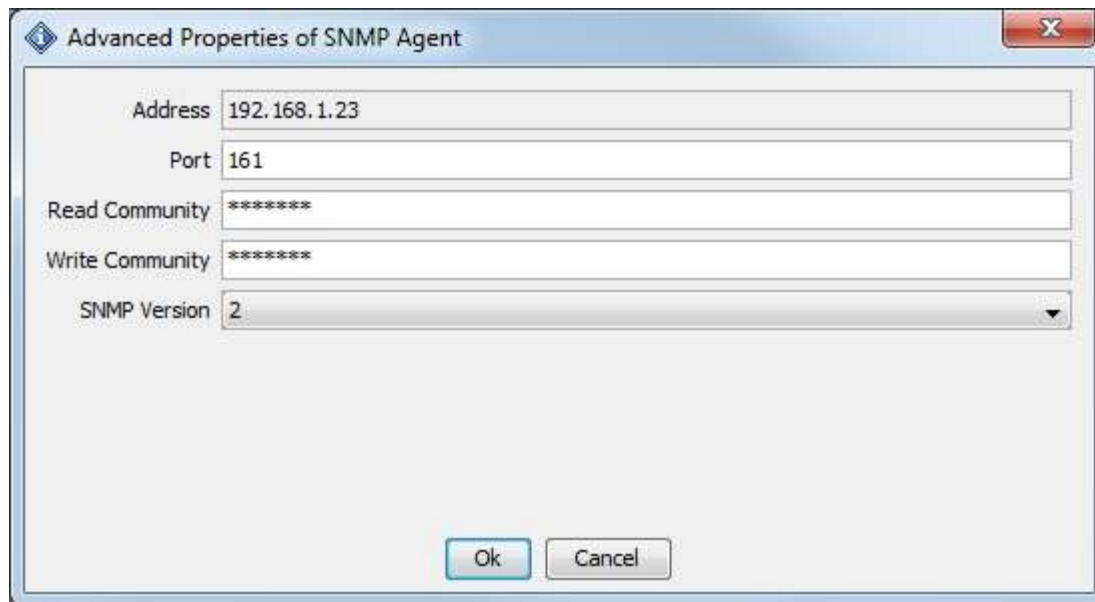
The MIB browser allows you to view the MIB variables in the Babel Buster. Before you can browse the MIB in a meaningful way, you need to load the MIB files that tell the browser what it needs to know about the BB3's MIB. There are one or more files that need to be loaded, and these are available under Documents and Links on the BB3-7302/MX-73 product page at csimn.com. Once you have loaded these files as illustrated below, you can view the tree structure of the MIB in the browser.



Enter the IP address of the Babel Buster BB3-7302/MX-73 in the Address window. In addition, click on Advanced... to set access parameters.



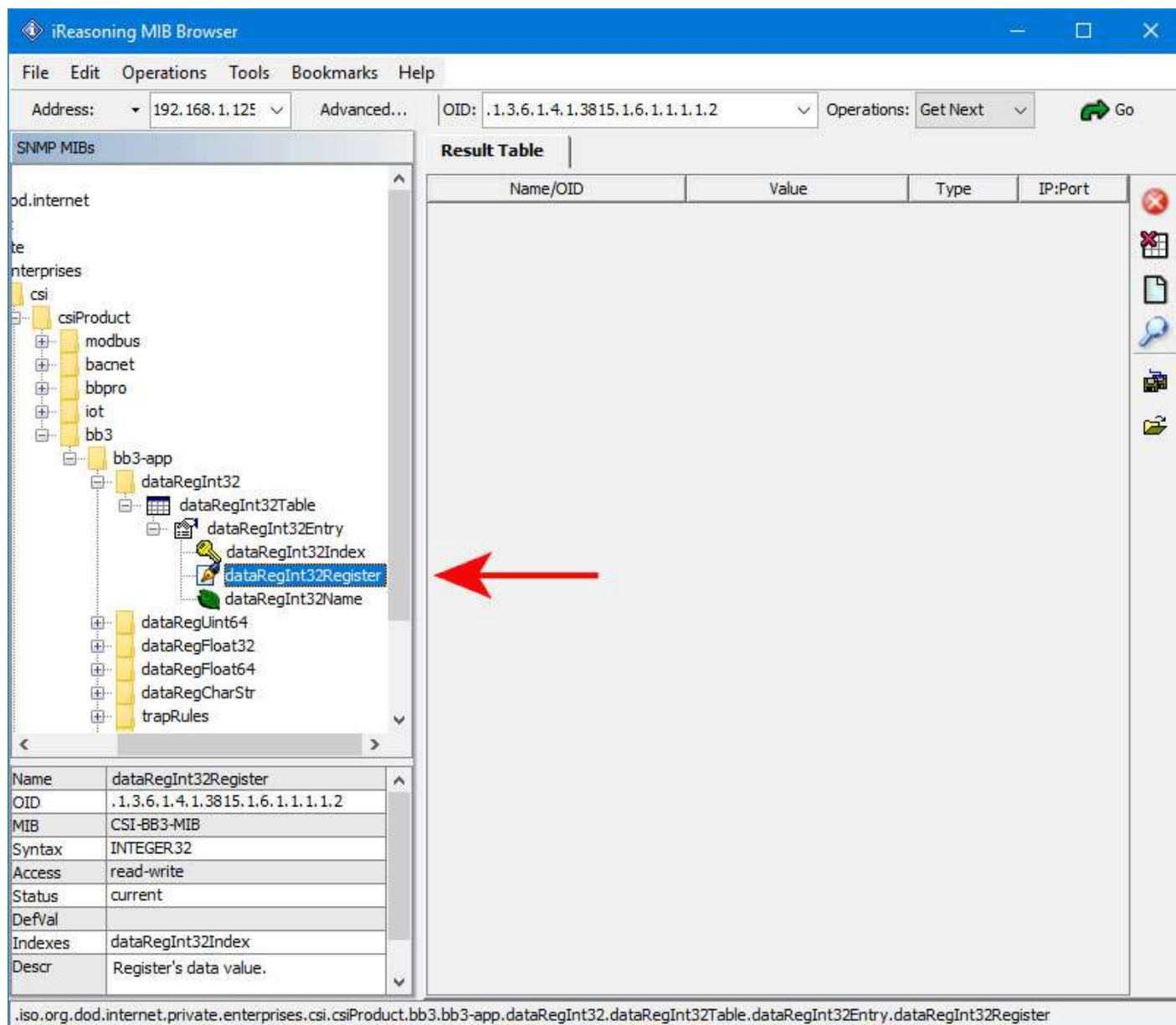
Click Advanced... to open the dialog that lets you enter the user name and community required for accessing the MIB in the Babel Buster.



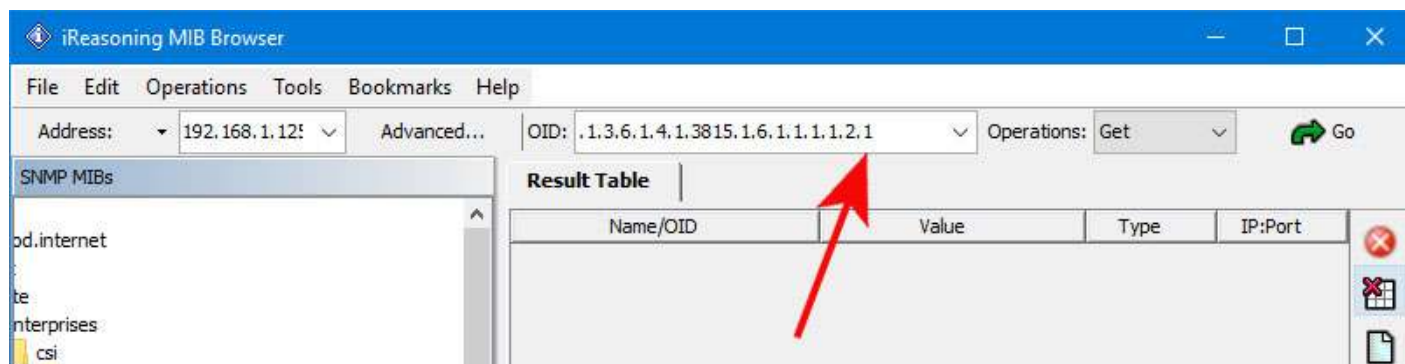
The image shows a Windows-style dialog box titled "Advanced Properties of SNMP Agent". It contains five input fields: "Address" with the value "192.168.1.23", "Port" with the value "161", "Read Community" with the value "*****", "Write Community" with the value "*****", and "SNMP Version" with a dropdown menu showing "2". At the bottom of the dialog are "Ok" and "Cancel" buttons.

Address	192.168.1.23
Port	161
Read Community	*****
Write Community	*****
SNMP Version	2

To read a MIB variable from the Babel Buster, start by selecting the object member of the data table. In this case, we are selecting the 32-Bit Integer branch of the MIB.

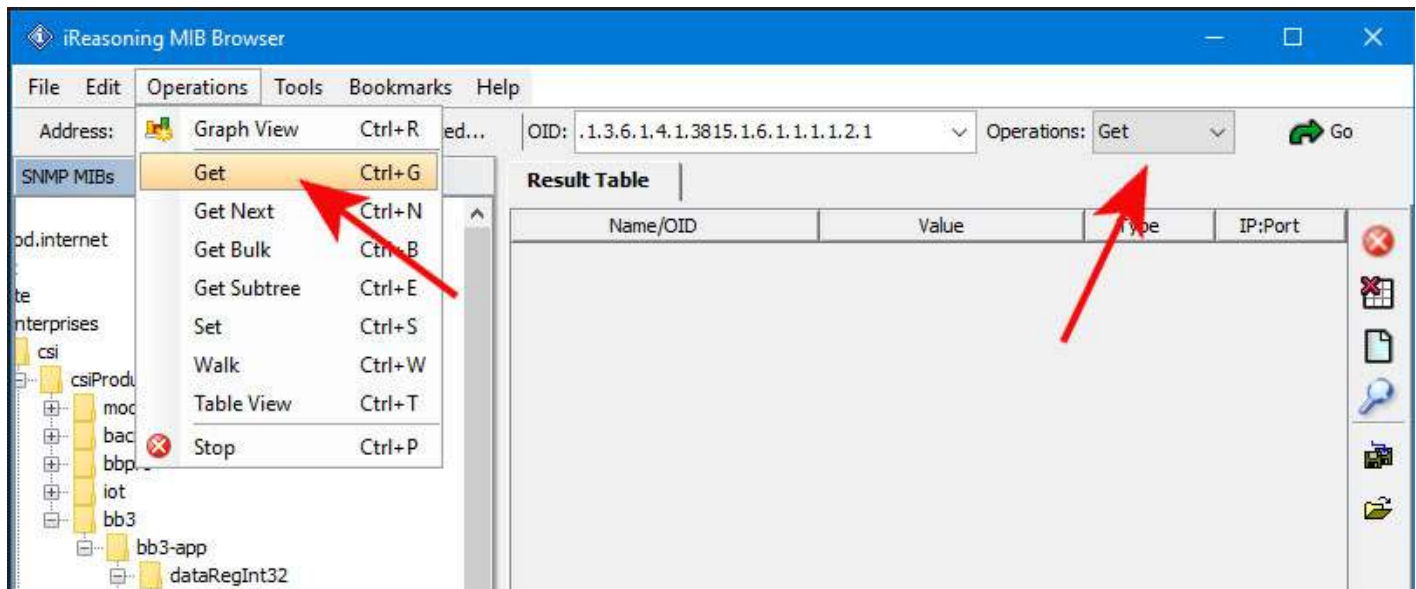


You will need to specify which row in the table you want to read. Do this by appending (by typing) a number to the OID that appeared in the OID window when you clicked on the table entry in the MIB tree view.

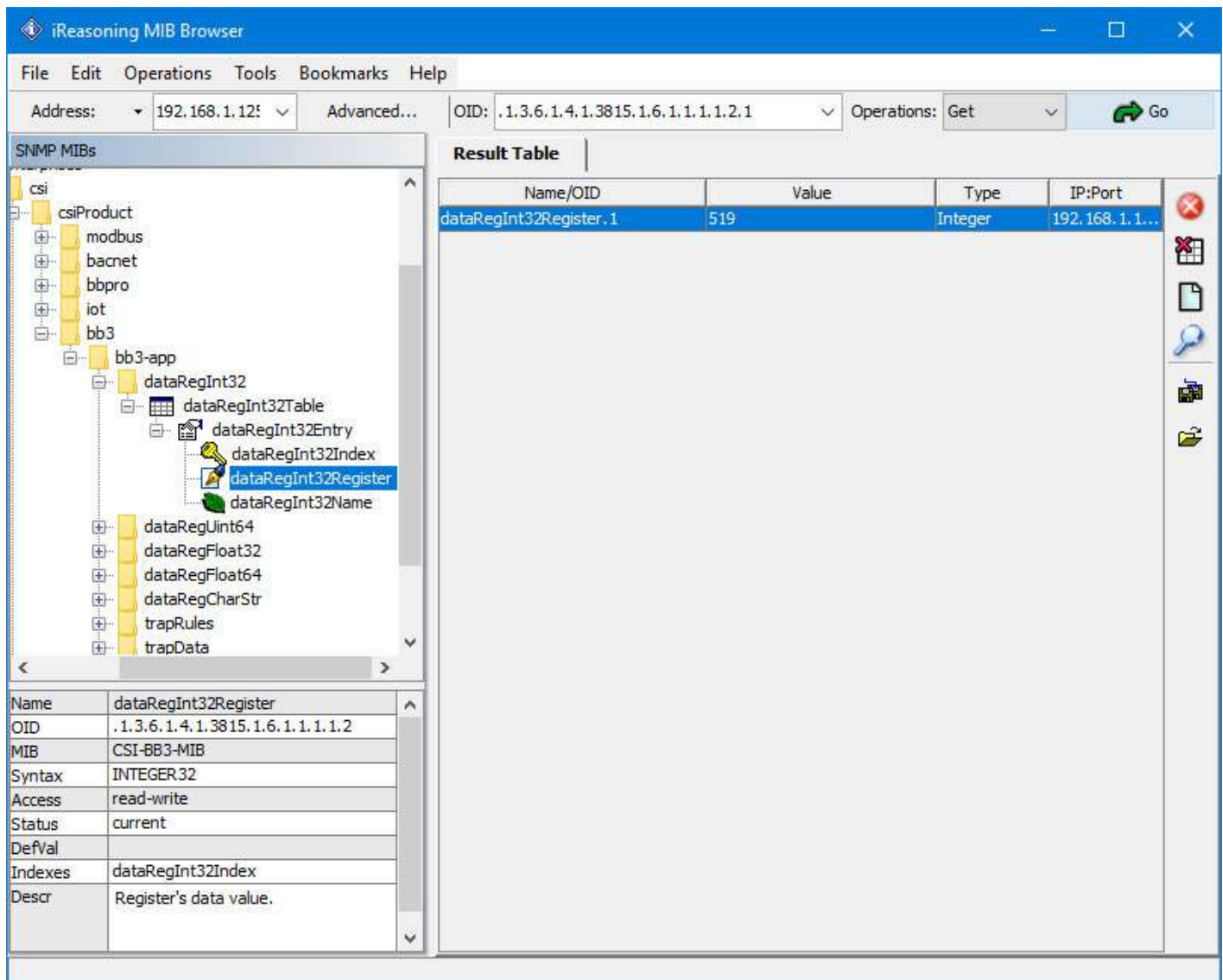


To cause the MIB browser to Get a value from the Babel Buster, select Get from the Operations menu. You can select Get from either menu, left or right, in the iReasoning browser. Once Get is selected on the right, you only need to click the Go button to

repeat the Get.



Upon successfully Getting a value, the display will appear as illustrated below.



If you attempt to Get a variable that does not exist, you will get the error message illustrated below. This will also happen if the variable has been added to the MIB but you forgot to click Reload SNMP after adding the local object to the MIB. This will also happen if you did not select the right variable from the MIB tree shown, or forgot to add the index to the end of the OID in the Object ID window.

The screenshot shows the iReasoning MIB Browser interface. The top bar includes a menu (File, Edit, Operations, Tools, Bookmarks, Help), an Address field (192.168.1.12), an Advanced... button, an OID field (.1.3.6.1.4.1.3815.1.6.1.1.1.2.88), an Operations dropdown (Get), and a Go button. The left pane displays a tree of SNMP MIBs, with the 'dataRegInt32Register' object selected under the 'bb3-app' subtree. The right pane shows a 'Result Table' with the following data:

Name/OID	Value	Type	IP:Port
dataRegInt32Register.1	519	Integer	192.168.1.1...
dataRegInt32Register.88	No Such Instance	NoSuchInst...	192.168.1.1...

Below the tree, a detailed view of the selected object is shown:

Name	dataRegInt32Register
OID	.1.3.6.1.4.1.3815.1.6.1.1.1.2
MIB	CSI-BB3-MIB
Syntax	INTEGER32
Access	read-write
Status	current
DefVal	
Indexes	dataRegInt32Index
Descr	Register's data value.

The status bar at the bottom displays the full OID path: .iso.org.dod.internet.private.enterprises.csi.csiProduct.bb3.bb3-app.dataRegInt32.dataRegInt32Table.dataRegInt32Entry.dataRegInt32Register.88

Getting our floating point value will appear as follows. The bytes "42 A4 83 12" translate to the floating point version of 82.255997. You can use Google to locate conversion tools to go from IEEE 754 hexadecimal representation (as shown in the browser) to decimal.

iReasoning MIB Browser

File Edit Operations Tools Bookmarks Help

Address: 192.168.1.12! Advanced... OID: .1.3.6.1.4.1.3815.1.6.1.3.1.1.2.1 Operations: Get Go

SNMP MIBs

enterprises

csi

csiProduct

modbus

bacnet

bbpro

iot

bb3

bb3-app

dataRegInt32

dataRegUInt64

dataRegFloat32

dataRegFloat32Table

dataRegFloat32Entry

dataRegFloat32Index

dataRegFloat32Register

dataRegFloat32Name

dataRegFloat64

dataRegCharStr

trapRules

Result Table

Name/OID	Value	Type	IP:Port
dataRegFloat32Register.1	0x42 A4 83 12	OctetString	192.168.1.1...

Name dataRegFloat32Register

OID .1.3.6.1.4.1.3815.1.6.1.3.1.1.2

MIB CSI-BB3-MIB

Syntax Float32TC (OCTET STRING) (SIZE(4))

Access read-write

Status current

DefVal

Indexes dataRegFloat32Index

Descr Register's data value.

.iso.org.dod.internet.private.enterprises.csi.csiProduct.bb3.bb3-app.dataRegFloat32.dataRegFloat32Table.dataRegFloat32Entry.dataRegFloat32Register.1



11. Configuring SNMP Trap Sender

11.1 SNMP Trap Destinations

The first step in sending traps is to tell Babel Buster where to send them. This is done on the Devices page of the Trap Sender. Traps or Notifications generated by this device will be sent to port 162 at the IP addresses provided here.

The screenshot shows the 'Trap Sender' configuration page in the Babel Buster 3 web interface. The top navigation bar includes 'Local Objects', 'BACnet', 'SNMP', and 'System'. Below this, a sub-navigation bar has 'Local MIB', 'Client Setup', 'Client Data', 'Trap Sender' (which is highlighted), and 'Trap Receiver'. The 'Devices' tab is selected on the left. The main content area shows 'Destination Device #' with a value of '1'. There are 'Update', '< Prev', and 'Next >' buttons. Below this, there are radio buttons for 'Use' with 'IPv4' selected and 'IPv6' unselected. There are also radio buttons for 'SNMP Version' with 'v1' selected and 'v2c' unselected. The 'IP Address' field contains '192.168.1.112' and the 'Community' field contains 'private'.

Traps may be sent as v1 or v2c. The same application varbinds will be included in either trap type, in addition to header and varbinds required for that trap type. (Note: We use the term "trap" generically here. The v2c "trap" is properly known as a Notification.)

To remove a destination, simply leave the IP address field blank. To add destinations, or change or remove existing entries, make the necessary changes here, then go to the File Manager page and save your configuration to the Boot configuration XML file. Then restart this device. The trap destinations are loaded one time prior to startup of the SNMP engine. To reload the trap destination table, you must restart the device.

The community string entered here will be included in traps sent to this destination.

11.2 SNMP Trap Triggers

Create a Trap Trigger for each trap that should be sent.

A trap must reference a variable in the local MIB. Therefore, before you can create a trap trigger, you must assign the local object of interest to a spot in the local MIB. (See "Configuring Gateway as an SNMP Server".) Once the local object has a home in the MIB, select the branch and MIB index in the trap trigger rule. Do not enter local object number here. Enter the MIB table index from your local MIB configuration pages.

The object found in the MIB branch, at the table index given, is displayed for reference as "Looking up...". To change this object, go to that MIB branch and table index, and change the MIB definition, or select a different branch and/or table index.

The screenshot shows the 'Trap Trigger' configuration window. At the top, there are three tabs: 'Devices', 'Trap Trigger' (selected), and 'Trap Summary'. Below the tabs, the 'Trap # 1' is displayed, along with the status 'Trigger rule presently tests FALSE'. To the right are 'Update', '< Prev', and 'Next >' buttons. The main configuration area has a dark background. It starts with 'Using MIB branch' followed by a dropdown menu showing 'Integer 32-Bit', and 'and table index: 1'. To the right, it says 'Looking up: AI 1: Analog Input 1'. Below this, the event is configured: 'Event is TRUE if the value is' followed by a dropdown showing 'Greater than', 'this value: 1000.000', and 'this object: None'. Further down, hysteresis is set to '0.00', 'this minimum On Time' to '0:00:00', and 'this minimum Off Time' to '0:00:00'. There are two checkboxes: 'Send if True' and 'Send if False', both with 'repeat 0 times' and 'Repeat Time: 0.0'. Below these are two text input fields: 'Message if True' with the value 'Data Value 1 is High' and 'Message if False' with the value 'Data Value 1 is Normal'. At the bottom, it shows '# Trap Trigger Rules Enabled: 2' and 'Insert' and 'Delete' buttons.

Once the variable of interest is selected, define the threshold. This will be a test such as "greater than" or "less than", etc. You can provide a fixed value for the threshold, or you may reference a local object that will hold a threshold that may change from time to time. If object is zero, then the fixed value will be used. If a non-zero object number is given, then the fixed value is disregarded. The trap trigger will be "true" when the MIB variable meets the criteria given.

Qualifications are optional, and enabled only when values are nonzero. How hysteresis is applied depends on the comparison. For a test that becomes true if greater than, the test will not return to false until the local object is less than the test value by a margin of at least this hysteresis value. If a test becomes true if less than, it will not return to false until the local object is greater than the test value by a margin of at least this hysteresis value.

Special test types: "Deviate from" will test against the value given, and use Hysteresis as the margin of deviation. This is effectively a "greater than or less than" test for deviation from a setpoint. "Changes by" will become true each time the given variable changes by the value given, and Hysteresis has no effect on this test. If "Changes by" references a value of zero, then this becomes a special test whereby the event is true any time something in the system updates the variable. A "changes by zero" should not be used when the variable is continuously read from a slave device since this will result in continuous traps.

You also have the option of specifying a minimum On and Off time. The "On" time means the rule must test true for this amount of time before the status will actually be set true and the trap will actually be sent. The "Off" time means the rule must test false for this amount of time before its status will actually be returned to false. Times are given in HH:MM:SS format (hours, minutes, seconds).

Check "Send if True" to send traps when the trap trigger rule meets all criteria for "true". Check "Send if False" if you would also like to send a trap when the trigger rule meets all criteria for "false".

Enter a repeat count if the trap should be repeated. If repeat count is zero, the trap will be sent one time. If repeat count is 1, the trap will be sent 2 times, and so on. The interval between traps will be the Repeat Time in seconds. Enter -1 for trap repeat count if the trap should simply repeat indefinitely at the Repeat Time interval. A repeat count of -1 for "Send if True" is acceptable. A repeat count of -1 for "Send if False" will be treated as no repeat since indefinite repeating of non-true events is ill-advised.

One of the varbinds in the trap message is an arbitrary ASCII character string, sent as an ASN Octet string. The "True" message will be sent when the trap event is true, and the "False" message will be sent when the trap event is false.

To delete the rule shown, click Delete. To insert a new rule before the rule displayed, click Insert. To add a rule to the end of the list, click Next when at the end of the list, enter new rule, and click Update. The number of rules enabled simply limits the scope of rule review so that you do not have to review a lot of unused rules. If the displayed rules are used up and you need more, increase the enabled number.

11.3 SNMP Trap Summary

The trap summary provides a page where you may see the present status of all of your trap trigger rules. Here we see the "true" status of the above trigger.

Devices		Trap Trigger		Trap Summary		Agent ID						
Showing 1 to 2 of 2										Update	< Prev	Next >
Trap #	Local MIB Variable		Local Object	Test Result	Local Value			Local Name				
1	Integer.1		AI 1	True	1105.000000			Analog Input 1				
2	---		None	False	0.000000			---				
1 Trap rule vars loaded				1 Trap data vars loaded				Reload SNMP				

11.4 SNMP Identity

The name, location, and contact listed above may be retrieved by the remote SNMP client.

The local community is the name that will be sent in Traps if no alternate community string is provided for the trap destination device. This name must also be used by the remote SNMP client to Get or Set this device. The name "public" is no longer accepted (unless you explicitly enter it here - not recommended).

Devices	Trap Trigger	Trap Summary	Agent ID
<div>System Name <input type="text" value="Babel Buster BB3-7302-V2C"/></div> <div>System Location <input type="text" value="St. Paul, Minnesota"/></div> <div>System Contact <input type="text" value="www.csimn.com"/></div> <div>Local Community <input type="text" value="private"/></div> <div>Update</div>			

11.5 Testing the SNMP Trap Sender

A variety of tools are available for browsing an SNMP MIB and receiving SNMP Traps. The tool used in the following examples is the iReasoning MIB Browser. Refer to the Tools section under Support at csimn.com for more information about SNMP tools.

The iReasoning MIB Browser allows you to browse the MIB, do table walks, etc., but of primary interest here, also allows you to test traps. When you open the iReasoning browser, under Tools, select Trap Receiver. The screen illustrated below shows the trap viewer after receiving the above trap from our Babel Buster.

The screenshot shows the iReasoning MIB Browser interface. The left pane displays the MIB Tree with 'iso.org.dod.internet' selected. The main pane shows the 'Trap Receiver' tab with a table of received traps. The table has columns: Description, Source, Time, and Severity. One trap is listed with Description 'trapOID: .iso.org.dod.internet.private.enterprises.csi.csiProduct.bb3...', Source '192.168.1.125', and Time '2020-08-19 09:13:40'. Below the table, the trap details are expanded, showing Source, Timestamp, SNMP Version, Trap OID, and Variable Bindings.

Description	Source	Time	Severity
trapOID: .iso.org.dod.internet.private.enterprises.csi.csiProduct.bb3...	192.168.1.125	2020-08-19 09:13:40	

Source: 192.168.1.125 **Timestamp:** 53 minutes 59 seconds **SNMP Version:** 2

Trap OID: .iso.org.dod.internet.private.enterprises.csi.csiProduct.bb3.bb3-app.bb3-traps.trapRuleStateChange

Variable Bindings:

Name: .iso.org.dod.internet.mgmt.mib-2.system.sysUpTime.0	Value: [TimeTicks] 53 minutes 59 seconds (323961)
Name: snmpTrapOID	Value: [OID] trapRuleStateChange
Name: .iso.org.dod.internet.private.enterprises.csi.csiProduct.bb3.bb3-app.trapData.trapDataTable.trapDataEntry.trapDataName.0	Value: [OctetString] Data Value 1
Name: .iso.org.dod.internet.private.enterprises.csi.csiProduct.bb3.bb3-app.trapData.trapDataTable.trapDataEntry.trapDataValue.0	Value: [OctetString] 1105.000000



12. Configuring Gateway as an SNMP Client

The SNMP Client is used to read and write (Get and Set) data in other SNMP devices. Data read from a remote SNMP device is stored in a local object when received. Data written to a remote SNMP device is taken from a local object when sent. The local objects are the same collection of objects that are accessible to BACnet or to other SNMP devices via the Babel Buster's own MIB.

12.1 SNMP Device Configuration

The list of other SNMP devices that will be accessed by the Babel Buster are entered on the Devices page under SNMP Client Setup.

The image shows the 'SNMP' tab in the Babel Buster 3 web interface. The top navigation bar includes 'Local Objects', 'BACnet', 'SNMP', and 'System'. Under 'SNMP', there are sub-tabs: 'Local MIB', 'Client Setup', 'Client Data', 'Trap Sender', and 'Trap Receiver'. The 'Client Setup' sub-tab is active. Below the navigation bar, there is a 'Devices' section with a table. The table has columns for 'Device #', 'Local Name', 'Use', 'IP Address', 'SNMP Community', 'Default Poll Period', 'Device Status', and 'Reset'. The first row shows 'Device # 1', 'Local Name: Server 1', 'Use: IPv4', 'IP Address: 192.168.1.87', 'SNMP Community: public', 'Default Poll Period: 5.0 Seconds', 'Device Status: 0', and a 'Reset' button. The 'Update' button is also visible.

Enter the IP address of the SNMP agent that the Babel Buster will send Get or Set requests to. You can enter either IPv4 or IPv6 addresses. Provide a local name for this device. This local name will appear in the device list when setting up Client Read Maps and Client Write Maps. Select whether v1 or v2 requests should be sent to this device (v3 is not supported by this model gateway). Your Get or Set request will not be honored by the remote SNMP device if you do not provide the correct community

string. Enter the community here. You will need to obtain that from the device you will be Getting or Setting.

Enter a default poll period. This sets the rate at which the Babel Buster will periodically Get or Set the variables identified in the Read and Write Maps.

12.2 SNMP Client Read Maps (Get)

Reading or Getting data from another SNMP device requires setting up a Read Map.

Local Objects		BACnet	SNMP	System		
Local MIB		Client Setup	Client Data	Trap Sender	Trap Receiver	
Devices	Client Read Map	Client Write Map	Table Walker			
1			Update	< Prev	Next >	
Map #	Remote SNMP OID	Remote Device	Data Hint	Local Object	Local Object Name	
1	1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.1	Server 1	Std. ASN	AI 1	Analog Input 1	
2	1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.2	Server 1	Std. ASN	AI 2	Analog Input 2	
3		None	Std. ASN	None	---	

To configure a Read Map, enter the OID of the variable you wish to Get, select the SNMP device from the device list, and provide a local object number where the received data should be placed. If the remote device will be providing floating point as RFC 6340, select that in the Data Hint list, otherwise leave the selection as "Std. ASN" which means standard ASN encoding.

Data hint helps the data parser figure out what the variable is. If ASN encoding is recognized as anything other than an octet string, the hint will be disregarded. If an octet string is found, then the parser needs to know if it should be treated as RFC 6340 floating point. If no hint is given (standard ASN), then the octet string will be treated as an ASCII character string, in which case ASCII to numeric conversion will be attempted automatically if the local object is numeric (string will simply be copied if result object is a character string type object). Note that standard, well-known ASN types are recognized as well as the NetSnmp ASN type for opaque float.

Devices		Client Read Map	Client Write Map	Table Walker		
		1		Update	< Prev	Next >
Map #	Remote SNMP OID	Remote Device	Data Hint	Local Object	Local Object Name	
1	1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.1	Server 1	Std. ASN	AI 1	Analog Input 1	
2	1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.2	Server 1	Std. ASN	AI 2	Analog Input 2	
3		None	Std. ASN	None	---	

Click on the Map number in the first column to access the expanded view of the Read Map where additional optional parameters may be entered.

Devices	Client Read Map	Client Write Map	Table Walker
Map # <input type="text" value="1"/> Update < Prev Next >			
Read OID <input type="text" value="1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.1"/> from <input type="text" value="Server 1"/> using data hint <input type="text" value="Std. ASN"/>			
Then apply scale: <input type="text" value="0.00"/> and offset: <input type="text" value="0.00"/>			
Save in local object <input type="text" value="AI 1"/> named <input type="text" value="Analog Input 1"/> Repeat this process every <input type="text" value="10.0"/> seconds.			
Apply this default value: <input type="text" value="0.00"/> after <input type="text" value="0"/> read failure(s).			
<input type="checkbox"/> Enable this map only when index object <input type="text" value="None"/> is set to a value of <input type="text" value="0"/>			
# Client Read Maps Enabled: <input type="text" value="3"/> Insert Delete			

The optional scale and offset may be used if the value being read is scaled, or you may simply use scale and offset to perform units conversions. If non-zero, the received data will be multiplied by scale, then added to offset, before being placed in the local object.

If the remote SNMP device should be read at some rate other than the default poll time given on the Devices page, enter that here.

If you wish to have a specific default value set in the local object in the event the Get fails, enter that default value and some non-zero number of read failures. The default value will be set after this number of failed attempts.

You have the option of making this Read Map conditional. If an index object number is provided and the Enable box is checked, then this read map will only be executed when the index object (local object) contains the value given. This allows multiple read maps to supply data to the same local object based on the value of the index object. It also allows reading to simply be suspended if a single read map supplies data to the local object. In a more sophisticated scenario, you could potentially suspend reading of the remote SNMP device if you know the device is powered down.

Map number simply tells you where you're at on the list of Read Maps. Click "next" and "prev" to scroll through the list. To advance directly to a specific map, enter the desired number in the "Map #" box, then click Update.

Delete will remove the map number shown in the "Map #" box. Insert will insert a new map before the map number shown, and is used for placing maps between existing maps. It is not necessary to use Insert to add maps to the bottom of the list or to define any map presently having zero for a source object or "none" for remote type.

Selecting "none" for remote type effectively deletes the map even though it will still appear in the list until deleted. Unused maps at the end of the list will always show none as the type. If you wish to prevent these from being displayed, reduce the number of maps enabled.

The number of maps enabled simply limits the scope of map review so that you do not have to review a lot of unused maps. If the displayed maps are used up and you need more, increase the enabled number.

12.3 SNMP Client Write Maps (Set)

Writing or Setting data to another SNMP device requires setting up a Write Map.

The screenshot shows the 'Client Write Map' configuration page. At the top, there are tabs for 'Local Objects', 'BACnet', 'SNMP', and 'System'. Under 'SNMP', there are sub-tabs for 'Local MIB', 'Client Setup', 'Client Data', 'Trap Sender', and 'Trap Receiver'. The 'Client Data' sub-tab is active, showing a table of write maps. The table has columns: Map #, Local Object, Remote SNMP OID, Remote Data Type, Remote Device, and Local Object Name. Map 1 is selected, showing 'AI 10' as the local object, '1.3.6.1.4.1.3815.1.2.2.1.1.1' as the remote OID, 'Integer 32-bit' as the data type, 'Server 1' as the device, and 'Analog Input 10' as the local object name. A dropdown menu is open for the 'Remote Data Type' column, showing options: 'Integer 32-bit', 'None', 'Integer 32-bit', 'Unsigned 64-Bit', 'Float 32-Bit', 'Float 64-Bit', 'Char String', 'Counter', 'Unsigned 32-Bit', and 'Opaque Float'. Below the table, there is a 'Quick Help' section with instructions on how to use the map configuration.

Map #	Local Object	Remote SNMP OID	Remote Data Type	Remote Device	Local Object Name
1	AI 10	1.3.6.1.4.1.3815.1.2.2.1.1.1	Integer 32-bit	Server 1	Analog Input 10
2	None		None	None	---

Quick Help

Write local objects out to remote SNMP OIDs. This page creates a map that can be used to write data to remote SNMP agents from data contained here. Click on map number to see more detail and insert/delete rules.

Rule number simply tells you where you're at on the list of OID map. To advance directly to a specific map, enter the desired number in the "Showing" box, then click Update.

Start by selecting the local object that will be the source of data to Set in the remote device. Enter the OID to write in that device. Select the data format that will be expected by that device at the OID given. Select a device from the device list. Only devices entered on the Devices tab will appear in this list.

Click on the Map number in the first column to access the expanded view of the Write Map where additional optional parameters may be entered.

The screenshot shows the expanded view of the 'Client Write Map' configuration for Map #1. The 'Map #' field is set to 1. The 'Read local object' is 'AI 10' named 'Analog Input 10'. The 'Write remote OID' section has two options: 'any time local object has changed by 0.00' (checked) or 'when 0.0 seconds have elapsed with no change'. The 'Otherwise write remote OID unconditionally. In any event, when writing remote OID, apply local object data as follows:' section has 'Apply scale: 0.00' and 'offset: 0.00'. The 'Write OID' is '1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.3' as 'Integer 32-bit' at 'Server 1'. The 'Repeat this process' section has 'at least' (checked) and 'no more than every 10.0 seconds'. The 'Enable this map only when index object None is set to a value of 0' is unchecked. At the bottom, the '# Client Write Maps Enabled' is 2, with 'Insert' and 'Delete' buttons.

Map # 1

Read local object AI 10 named Analog Input 10

Write remote OID ☐ any time local object has changed by 0.00 or ☐ when 0.0 seconds have elapsed with no change.

Otherwise write remote OID unconditionally. In any event, when writing remote OID, apply local object data as follows:

Apply scale: 0.00 and offset: 0.00

Write OID 1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.3 as Integer 32-bit at Server 1

Repeat this process ☒ at least ☐ no more than every 10.0 seconds.

☐ Enable this map only when index object None is set to a value of 0

Client Write Maps Enabled: 2

Insert Delete

The local object data may be written to the remote SNMP device periodically, or when it changes, or both. To send upon change (send on delta), check the first box and enter the amount by which the local object must change before being written to the remote device. To guarantee that the remote OID will be written at least occasionally even if the data does not change, check the second box and enter some amount of

time. This time period will be referred to as the "maximum quiet time".

Data from the local object may be manipulated before being written to the remote OID. The local data is first multiplied by the scale factor. The offset is then added to it.

Enter the OID to write in that device. Select the data format that will be expected by the selected device at the OID given. Select a device from the device list. Only devices entered on the Devices tab will appear in this list.

The repeat time may determine how often the remote OID will be written. If send on delta and maximum quiet time are not checked above, clicking the "at least" button will establish a periodic update time. If send on delta is used and you wish to limit the network traffic in the event changes are frequent, click the "no more than" button and enter the minimum time that should elapse before another write to the remote device. It is valid to select "no more than every 0.0 seconds" if you want all changes to be sent, but no periodic writes.

You have the option of making this Write Map conditional. If an index object number is provided and the Enable box is checked, then this write map will only be executed when the index object (local object) contains the value given. This allows multiple write maps to supply data to the same remote OID based on the value of the local index object. It also allows writing to simply be suspended if a single write map supplies data to the remote OID. In a more sophisticated scenario, you could potentially suspend writing of the remote device if you know the device is powered down.

Delete will remove the map number shown in the "Map #" box. Insert will insert a new map before the map number shown, and is used for placing maps between existing maps. It is not necessary to use Insert to add maps to the bottom of the list or to define any map presently having zero for a source object or "none" for remote type.

Selecting "none" for remote type effectively deletes the map even though it will still appear in the list until deleted. Unused maps at the end of the list will always show none as the type. If you wish to prevent these from being displayed, reduce the number of maps enabled.

12.4 SNMP Client Data Displayed by Agent

The Client Data page shows the list of local objects mapped to a remote SNMP device.

Babel Buster[®] 3
BACNET
ROUTER & GATEWAY
MODEL BB3-7302-V2C

CONTROL SOLUTIONS MINNESOTA

Local Objects BACnet **SNMP** System

Local MIB Client Setup **Client Data** Trap Sender Trap Receiver

Client Data Errors: Read Maps Errors: Write Maps Errors: Table Walk

SNMP Device #1 Showing 1 to 3 of 3 Update < Prev Next >

Remote OID	Object Name	Local Object	Object Data	Time since Last update
(R) 1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.1	Analog Input 1	AI 1	12.000000	2.980
(R) 1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.2	Analog Input 2	AI 2	375.000000	2.880
(W) 1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.3	Analog Input 10	AI 10	0.000000	2.790

SNMP Device # 1 < Prev Dev Next Dev >

Click the Next Dev or Prev Dev buttons to go to the next or previous SNMP device, or simply enter a number in the SNMP Device # window and click Update. Maps for that device will now be displayed. In addition to a summary of the map (both read and write maps are shown), the time since last update is displayed. This time should generally be less than the poll time. If the last update time is large, it may mean there is an error preventing the update.

12.5 Supported Data Formats

SNMP does not have a universally accepted representation for floating point. The one universally known data type is INTEGER. A commonly recommended means of transmitting floating point data is either as a scaled integer or as an ASCII character string. A well known but application specific implementation (NetSNMP) uses ASN OPAQUE FLOAT. There does exist an RFC 6340 for representation of floating point. Both the NetSNMP and RFC 6340 versions are based on IEEE 754 encoding. The "Float 32-bit" and "Float 64-bit" data types in the Babel Buster's own MIB refer to RFC 6340 encoding.

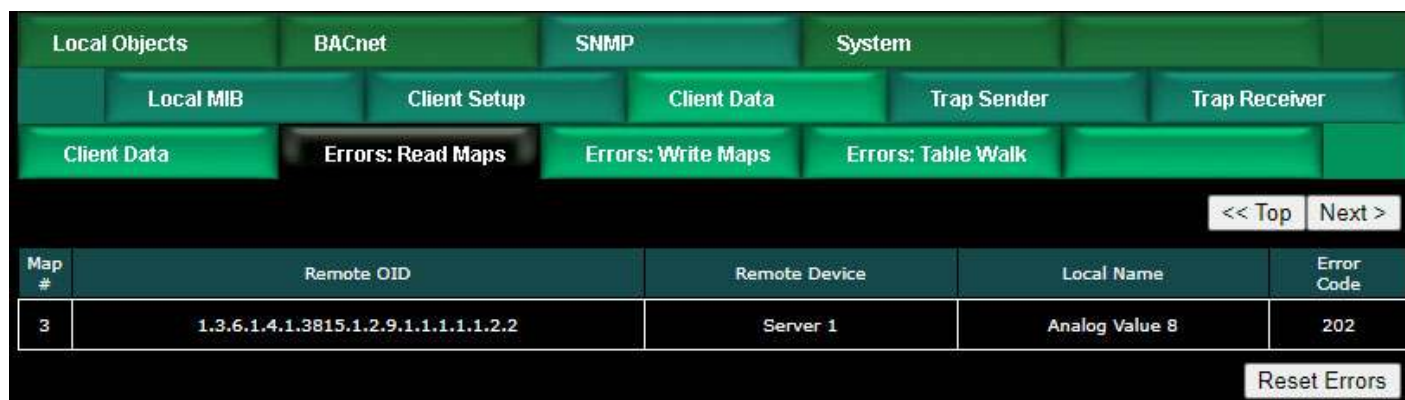
Data types recognized by Babel Buster when Getting or Setting SNMP data in other devices are listed in the table below. Specifically, the data types are encoded with ASN types as follows:

Integer 32-Bit	INTEGER	ASN_INTEGER
Unsigned 64-Bit	COUNTER64	(ASN_APPLICATION 6)
Float 32-bit	OCTET STRING	ASN_OCTET_STR (length 4)
Float 64-bit	OCTET STRING	ASN_OCTET_STR (length 8)
Char String	OCTET STRING	ASN_OCTET_STR (length variable)
Counter	COUNTER	(ASN_APPLICATION 1)

Unsigned 32-bit	UNSIGNED	(ASN_APPLICATION 2)
Float-Opaque	OPAQUE FLOAT	(ASN_APPLICATION 8)

12.6 SNMP Errors

The errors pages will show Read Maps and Write Maps that currently have an error status.



Error codes that may be returned by the SNMP client are as follows:

Standard SNMP error codes returned by remote Agent:

- 1 = SNMP_ERROR_tooBig
- 2 = SNMP_ERROR_noSuchName
- 3 = SNMP_ERROR_badValue
- 4 = SNMP_ERROR_readOnly
- 5 = SNMP_ERROR_genErr
- 6 = SNMP_ERROR_noAccess
- 7 = SNMP_ERROR_wrongType
- 8 = SNMP_ERROR_wrongLength
- 9 = SNMP_ERROR_wrongEncoding
- 10 = SNMP_ERROR_wrongValue
- 11 = SNMP_ERROR_noCreation
- 12 = SNMP_ERROR_inconsistentValue
- 13 = SNMP_ERROR_resourceUnavailable
- 14 = SNMP_ERROR_commitFailed
- 15 = SNMP_ERROR_undoFailed
- 16 = SNMP_ERROR_authorizationError
- 17 = SNMP_ERROR_notWritable
- 18 = SNMP_ERROR_inconsistentName

Client generated errors:

- 201 = No response from remote Agent (server)
- 202 = No such name (implicit)
- 203 = Unable to interpret application data
- 204 = Reply does not match request
- 205 = There is a problem with the map configuration
- 206 = Unable to build or parse SNMP PDU
- 207 = Table walk came up short of expected count

If the error code indicates no response, the device status (Client Setup Devices page) will provide an additional indication of a connection related error.

The screenshot shows a web interface for configuring an SNMP client. At the top, there are tabs: 'Devices' (selected), 'Client Read Map', 'Client Write Map', and 'Table Walker'. Below the tabs, there is a 'Device #' field with the value '1' and buttons for 'Update', '< Prev', and 'Next >'. The main configuration area has the following fields: 'Local Name' (APC UPS), 'Use' (radio buttons for IPv4 and IPv6, with IPv4 selected), 'SNMP Version' (radio buttons for v1 and v2c, with v1 selected), 'IP Address' (192.168.1.19), 'SNMP Community' (public), and 'Default Poll Period' (2.0 Seconds). On the right side, there is a 'Device Status' field showing '11' and a 'Reset' button.

Device status may indicate the following application level error conditions:

10 = could not bind socket

11 = response timed out (may mean IP address given is not reachable)

Device status may also indicate error codes from 80 and up are IP stack errors. Since SNMP uses UDP, which is connection-less, the only likely errors are:

104 = Connection reset by peer

111 = Connection refused

113 = Connection aborted

114 = Network is unreachable

115 = Network interface not configured

116 = Connection timed out

118 = Host is unreachable

125 = Address not available



13. Configuring SNMP Table Walker

13.1 Table Walk Methods

Visualize the SNMP table as a spread sheet. Although there can technically be any number of dimensions to an SNMP table, the table walker here is limited to 2-dimensional tables, just like a spread sheet. The table consists of some number of columns each of which are identified by an OID. The table contains any number of rows, which are identified by a table index. The table index, or row number, simply appears as the last field in the OID (or in other words, append the row number to the OID that identifies the column).

The Babel Buster table walker will scan the table, picking out one column per walk rule. For each row in the table, that column will be retrieved and its data placed into the local object provided. The first data value retrieved will be placed into the starting local object, and the walk will continue until the number of objects specified by "count" are filled.

The table walk method will default to "Normal" if you use only the quick entry of a simple table walk rule. This means you enter a starting OID, select a device, enter a starting object and number of local objects to fill with table content, and set a poll rate (meaning how often to periodically walk the table).

Rule #	Table Name (OID)	Device	Start Object	Count	Poll Rate (S)
1	1.3.6.1.2.1.33.1.6.2.1.*(2).*	APC UPS	BI 2	24	15
2		None	None	0	0

To select any table walk method other than Normal, you need to click on the table

walk rule number in the first column, and select the method on the expanded view of the walk rule.

The screenshot shows the 'Table Walker' tab in a configuration tool. At the top, there are tabs for 'Devices', 'Client Read Map', 'Client Write Map', and 'Table Walker'. Below the tabs, there is a 'Rule #' field with the value '1'. To the right of this field are 'Update', '< Prev', and 'Next >' buttons. The main configuration area has several fields: 'Walk table at this name (OID):' with the value '1.3.6.1.2.1.33.1.6.2.1.*(2).*', 'using method:' with a dropdown menu showing 'Index' selected, 'From device at:' with a dropdown menu showing 'APC UPS', 'using data hint:' with a dropdown menu showing 'Std.ASN', 'Save in local objects starting at:' with the value 'BI 2', 'saving up to this count:' with the value '24', 'Repeat this process every:' with the value '15' and 'seconds.', 'Set optional object timeout to:' with the value '20' and 'seconds.', and a checkbox 'Enable this table walk only when index object' with a dropdown menu showing 'None' and 'is set to a value of' with the value '0'. At the bottom, there is a field '# Table WalkRules Enabled:' with the value '2', and 'Insert' and 'Delete' buttons.

The table walking process can become complicated by the fact that tables are allowed to have missing rows, or rows may have intermittent missing columns. In some applications, rows will be populated only temporarily and then they will disappear. The table walk method identified as "Normal" expects the given column to exist in all rows, and at all times, through the range of rows defined by the starting OID and the count. The remaining methods accommodate the other complications permitted by SNMP.

Method "**Normal**" will simply produce a 1 to 1 correlation between table entries and object numbers, placing successive values in successive objects. Data will be interpreted according to the data hint if an octet string is returned, otherwise the ASN encoding will take precedence. If the Get-Next sequence fails to return enough OIDs to fill the 'count' criteria, an error code is set for the device indicating that the table came up short on data.

Method "**Sparse**" is the same as Normal, except missing OIDs in the sequence is anticipated, and the corresponding local objects in the sequence are skipped over if the respective OID is not included in the Get-Next sequence. No error is flagged for the table being short on data.

Method "**Wildcard**" allows wildcard fields in the table OID. The walk does not care about order of OIDs returned by Get-Next as long as they match the OID given after discounting wildcard fields. No attempt is made to sequentially pair OIDs with objects. The next 'count' OIDs that successfully match the OID with wildcards will fill the next 'count' of objects beginning with the starting local object number. The OID sent out in the first Get-Next request will have zero in any wildcard fields, and each successive Get-Next will send out the OID from the response to the previous Get-Next.

Method "**Index**" will walk the table, but expect to find that values are OIDs. In other words, the table name is an OID, but the contents of the variable at that name will be another OID. The result is that the OID index (last field of OID) from the value will be used as the offset to calculate which object number is to be affected, and that object will be set to 1 indicating this OID is present in the table. This seemingly odd means of table walking is required in order to translate the alarm table from RFC 1628 for UPS

systems into indexable BACnet objects that indicate the presence or absence of alarms defined in RFC 1628. (Note that the alarm table in RFC 1628 is "sparse" meaning its table entries are only present if an alarm is present, and the table is empty if there are no alarms. One cannot simply query OIDs to determine presence of an alarm. Alarms are implied by presence of a table entry that only exists while the alarm is present. Furthermore, the alarm table is simply a circular buffer of alarm entries, and the table index means nothing.)

The object set to 1 by the Index method will not be reset to zero by reading anything from the table. The object is set by finding something in the table, and not finding anything in the table will have no effect. Therefore, once set, the object will remain set indefinitely unless the table rule timeout is used. The object will be reset to zero when the timer expires. The timer is reset every time the table walk writes to this object. But this periodic setting of the object will stop if the previously found table entry is no longer present, and then the timeout will clear or reset the object (e.g., change a Binary object from Active to Inactive).

13.2 Configuring Table Walk Rules

The table walk rules apply to a specific SNMP device whose table is to be walked. The first step in creating a table walk is to enter the device information on the Devices tab. The same set of devices are used for SNMP client read and write rules.

Local Data		BACnet		SNMP		System	
Local MIB		Client Setup		Client Data		Trap Sender	
Devices		Client Read Map		Client Write Map		Table Walker	
Showing 1 to 2 of 2				Update		< Prev Next >	
Rule #	Table Name (OID)	Device	Start Reg #	Count	Poll Rate (S)		
1	1.3.6.1.2.1.33.1.6.2.1.*(2).*	APC UPS	103	24	30		
2		None	0	0	0		

Enter the IP address of the SNMP device, and provide a local name. This local name is only used to populate the device list on the rules pages. The SNMP version will be either v1 or v2c for this gateway (v3 is not supported here - v3 requires a different model). This determines which version is indicated in the Get-Next requests generated by the table walker. The community should match whatever the SNMP device to be queried is expecting to see. The default poll rate is used whenever the rate in the individual rule is left at zero.

Next, proceed to the Table Walker page. The page that opens initially is a tabular list of multiple rules. You can enter everything necessary for a "Normal" method table walk using just this tabular form. However, if you will be using wildcards in the OID or any method other than Normal, you must click the rule number in the first column and set up the rule using the expanded form. Upon clicking a rule number, the expanded form of the rule is displayed. To return to the tabular list, click on the Table Walker tab again.

Local Objects		BACnet		SNMP		System			
Local MIB		Client Setup		Client Data		Trap Sender		Trap Receiver	
Devices		Client Read Map		Client Write Map		Table Walker			

Showing 1 to 2 of 2

Rule #	Table Name (OID)	Device	Start Object	Count	Poll Rate (S)
1	1.3.6.1.2.1.33.1.6.2.1.*(2).*	APC UPS ▼	BI 2	24	15
2		None ▼	None	0	0

The parameters for a table walk rule are as follows. Only the table name, device, starting object, count, and poll rate are entered on the tabular list of rules. To enter the additional parameters or use a method other than "Normal", the expanded view must be used.

Table Name or OID: The walk will begin at the variable name or OID given. It is not necessary to start at the beginning of a table. To walk a section of the table, enter an OID whose last field is the desired first index, minus one. The count will limit the scope of the walk.

The table name OID includes optional wildcard fields when the walk method is Index or Mask. For example, to walk the alarm table of a UPS using RFC 1628, the OID would be 1.3.6.1.2.1.33.1.6.2.1.*(2).*. The *(2) means allow any value in the second to last field but only act on the value when this field is 2. The last asterisk means disregard the last field entirely (which for RFC 1628 increments with each new alarm until reaching some rollover point, then starts back at 1 again).

Method: Select method as defined above in section 13.1.

Device: Select an SNMP device from the list created in the Devices page.

Data Hint: Data hint provides instruction for interpretation of data in the event the value is an Octet String. If the data is an ASN type that is clearly recognizable, the ASN encoding will take precedence. If the Octet String should be interpreted per RFC 6340 for floating point, select RFC 6340. Otherwise, leave the "Std. ASN" selection as is. An Octet String will be treated as a character string if the destination local object is defined as a character string. Otherwise, an attempt will be made to convert the Octet String as if it were an ASCII representation of some number. If the octet string contains no numeric information, the result will be zero.

Starting Local Object and Count: The "local objects starting at #" is the first BACnet object to be filled by this table walk, and "count" is the number of objects in the sequence that are to be affected. Count is a number of object instances. The count also determines how many OIDs in the table will be read since there is a 1 to 1 correspondence between table entries and local object values saved.

Poll Rate (Repeat...): This sets the rate at which the table should be walked. The amount of time to wait between table walks is entered. Be prudent with poll rate - this will generate a lot of network traffic.

Timeout: The object will be set to zero after this amount of time has expired with no update from the table walk. This only applies when sparse tables are walked (e.g. RFC 1628 alarm table).

Enable Index Object: You have the option of enabling this rule only when a selected object contains a given value. Any local object may be used as the index object. As the name implies, you could have the same local objects contain different values based on different rules as indexed by the index object.

Click Update to object your changes. Don't forget to save all changes in the configuration file when done. To insert a rule before an existing rule, enter the existing rule number in the "Rule #" window at the top, and click Insert. To delete an existing rule, enter the rule number in the "Rule #" window and click Delete.

Table Walk Example 1:

The screenshot shows the 'Table Walker' configuration window. At the top, there are tabs for 'Devices', 'Client Read Map', 'Client Write Map', and 'Table Walker'. Below the tabs, the 'Rule #' is set to 1. To the right of the rule number are 'Update', '< Prev', and 'Next >' buttons. The main configuration area includes: 'Walk table at this name (OID):' with the value '1.3.6.1.2.1.33.1.6.2.1.*(2).*' and 'using method:' set to 'Index'. 'From device at:' is set to 'APC UPS' and 'using data hint:' is set to 'Std.ASN'. 'Save in local objects starting at:' is set to 'BI 2' and 'saving up to this count:' is set to '24'. 'Repeat this process every' is set to '15' seconds, and 'Set optional object timeout to' is set to '20' seconds. There is a checkbox labeled 'Enable this table walk only when index object' which is currently unchecked, with 'None' selected for the index object and '0' for the value. At the bottom, '# Table WalkRules Enabled:' is set to '2'. To the right of this are 'Insert' and 'Delete' buttons.

The table walk rule illustrated above will walk the alarm table in a UPS that implements RFC 1628. Using the Index method, Binary Input objects illustrated below will be set to "active" when the corresponding alarm condition is present. RFC 1628 defines 24 "well known" alarm conditions. Alarm condition 2 is defined to mean "UPS is on battery". The screen shot below illustrates the UPS being on battery, with no other alarms currently present.

Babel Buster Pro V230
BACnet-SNMP
NETWORK GATEWAY

CONTROL SOLUTIONS, INC.
MINNESOTA

Local Objects | BACnet | SNMP | System

Analog | Binary | Multi-State | Actions

Input Objects | Output Objects | Value Objects

Binary Input Objects Showing objects from 1 Update < Prev Next >

Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Text
1	upsOnBattery UPS on battery trap detected	N	Active	0	0,0,0,0	UPS is on battery
2	upsAlarmBatteryBad One or more batteries need replacement	N	Inactive	0	0,0,0,0	Status Normal
3	upsAlarmOnBattery Indicates when UPS is on battery	N	Active	0	0,0,0,0	UPS on Battery Power
4	upsAlarmLowBattery Remaining battery time less than limit	N	Inactive	0	0,0,0,0	Status Normal
5	upsAlarmDepletedBattery UPS will be unable to sustain load	N	Inactive	0	0,0,0,0	Status Normal
6	upsAlarmTempBad Temperature out of tolerance	N	Inactive	0	0,0,0,0	Status Normal
7	upsAlarmInputBad An input condition is out of tolerance	N	Inactive	0	0,0,0,0	Status Normal

The Index method of table walk will set objects corresponding to OIDs found that reference the "well known alarms" defined in RFC 1628. Objects are skipped if no corresponding OID is found in the table. That means the Index walk will set objects but never clear them. To get the objects to reset after being set to indicate an alarm, the timeout feature of the table walk rule must be used.

As illustrated above, the timeout is set to 20 seconds. The table walk in this case is set to repeat every 15 seconds. If the alarm is still present, it will be set once again. The timer is reset every time the table walk writes to this object. But this periodic setting of the object will stop if the alarm is no longer present, and then the 20 second timeout will clear or reset the object (change a Binary object from Active to Inactive).

Table Walk Example 2:

The next example is a simple table walk using the Normal method, included here for completeness of discussion. The real power of the Babel Buster is in its ability to walk the types of tables that require more complex methods. The Normal table walk illustrated here does not need to be a table walk at all. It could just as easily be a set of SNMP client read map rules. The client read rules would periodically Get the values named. Likewise, the Normal table walk periodically Gets the values named. The problem with relying solely upon periodic polling is that in applications like RFC 1628, alarm OIDs only exist while the alarm is active and any attempt to do a direct Get at other times results in an error. Such a problem is solved by the other methods

available for table walking in the Babel Buster.

To illustrate the Normal table walk, we will get values from the contiguous full table provided by another gateway, the BB2-6010. A screen shot of its MIB View shows the OIDs we will retrieve with our table walk.

This page displays data as presently found in the local registers maintained by this device.

Showing registers from

Local Register #	Register Name	Update	Register Data	Register Type	SNMP MIB OID
00001	Level 1 Control	<input type="checkbox"/>	17	Integer	1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.1
00002	Level 2 Control	<input type="checkbox"/>	39	Integer	1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.2
00003	Level 3 Control	<input type="checkbox"/>	68	Integer	1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.3
00004		<input type="checkbox"/>	0	Integer	1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.4

The screen shot of the walk rule list shows the previous example plus this example added as a second rule. Because this is a simple "Normal" table walk, everything needed can be entered on the tabular rule list. The first OID to be read is given as the table name. Select the device, starting local object and count, and poll rate. That is all that is needed. (Note: The device in this example has already been set up over on the devices page, first tab.)

Showing to 3 of 3

Rule #	Table Name (OID)	Device	Start Object	Count	Poll Rate (S)
1	1.3.6.1.2.1.33.1.6.2.1.*(2).*	APC UPS	BI 2	24	15
2	1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.1	BB2-6010	AV 5	3	15
3		None	None	0	0

If you were to click the rule number (2) to look at the expanded form of this rule, here is what you would see. No additional entries are required to make this table walk functional.

Rule # **2** Update < Prev Next >

Walk table at this name (OID): **1.3.6.1.4.1.3815.1.2.2.1.1.1.1.2.1** using method: **Normal**

From device at: **BB2-6010** using data hint: **Std.ASN**

Save in local objects starting at **AV 5** saving up to this count: **3**

Repeat this process every **15** seconds. Set optional object timeout to **0** seconds.

☐ Enable this table walk only when index object **None** is set to a value of **0**

Table WalkRules Enabled: **3** Insert Delete

Given the screen shot of the BB2-6010 MIB View, here is what we expect to see in the Babel Buster BB3-7302/MX-73 when the table walk starts running.

Local Objects	BACnet	SNMP	System
Analog	Binary	Multi-State	Actions
Input Objects	Output Objects	Value Objects	

Analog Value Objects Showing objects from **1** Update < Prev Next >

Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Units
<u>1</u>	Analog Value 1	N	0.000000	0	0,0,0,0	no_units
<u>2</u>	Analog Value 2	N	0.000000	0	0,0,0,0	no_units
<u>3</u>	Analog Value 3	N	0.000000	0	0,0,0,0	no_units
<u>4</u>	Analog Value 4	N	0.000000	0	0,0,0,0	no_units
<u>5</u>	Analog Value 5	N	17.000000	0	0,0,0,0	no_units
<u>6</u>	Analog Value 6	N	39.000000	0	0,0,0,0	no_units
<u>7</u>	Analog Value 7	N	68.000000	0	0,0,0,0	no_units
<u>8</u>	Analog Value 8	N	0.000000	0	0,0,0,0	no_units
<u>9</u>	Analog Value 9	N	0.000000	0	0,0,0,0	no_units
<u>10</u>	Analog Value 10	N	0.000000	0	0,0,0,0	no_units

Table Walk Example 3:

The following example elaborates on the RFC 1628 alarm table walk originally illustrated above. We are now going to walk the alarm table conditionally. Namely, we will only walk the table when we know there are alarms in the table. There is a variable in the RFC 1628 MIB that tells us how many alarm entries are currently in the table. This is found at the OID illustrated below, and we have set up an SNMP client read map to retrieve that count, and place it in local object AI 14.

Our definition of AI 14 looks like this:

Local Objects BACnet SNMP System

Analog Binary Multi-State Actions

Input Objects Output Objects Value Objects

Analog Input # 14 Update < Prev Next >

Reliability: 0 Status: 0,0,0,0 Device Link: SNMP_R16 Out of Service: ☐ Deconfigure: ☐

Object name upsAlarmCount Force ☐ Present Value 0.000000

Description Count of alarms in alarm table

COV increment: 0.000000 Units: no_units

The SNMP Client Read Map looks like this:

Local Objects BACnet SNMP System

Local MIB Client Setup Client Data Trap Sender Trap Receiver

Devices Client Read Map Client Write Map Table Walker

Showing 3 to 17 of 17 Update < Prev Next >

Map #	Remote SNMP OID	Remote Device	Data Hint	Local Object	Local Object Name
16	1.3.6.1.2.1.33.1.6.1.0	APC UPS	Std. ASN	AI 14	upsAlarmCount

We will create a Binary Input object to hold the result of the rule count test, as follows:

Local Objects BACnet SNMP System

Analog Binary Multi-State Actions

Input Objects Output Objects Value Objects

Binary Input # 26 Update < Prev Next >

Reliability: 0 Status: 0,0,0,0 Device Link: --- Out of Service: ☐ Deconfigure: ☐

Object name upsAlarmsPresent Force ☐ Present Value Inactive

Description Derived indication if alarms present

Active Text: Alarms present Inactive Text: Alarm table empty

The index object in the table walk rule wants to see a specific number. Since there may be zero to many alarms, we need a way of simply reducing that to an effective "yes or no" to the walk rule. We accomplish this by simply testing to see if the alarm count is greater than zero by using a Calculate rule. The result of the test will be 0 (false) or 1 (true) with the result placed in Binary Input 26.

Local Objects		BACnet		SNMP		System	
Analog		Binary		Multi-State		Actions	
Calculate		Copy					

Showing 1 to 2 of 2 Update < Prev Next >

Rule #	Perform Operation	Using Object	And/Through	This Object	Place Result in Object
1	test > 0	AI 14	and	None	BI 26
2	none	None	and	None	None

Rules Enabled: 2 Insert Delete

Now that we have a object whose value will be 1 (active) only when one or more alarms are in the RFC 1628 alarm table, we can enable the index object in the table walk rule as illustrated below. We have now reduced network traffic by only walking the alarm table when we know alarms are present.

Devices	Client Read Map	Client Write Map	Table Walker
---------	-----------------	------------------	--------------

Rule # 1 Update < Prev Next >

Walk table at this name (OID): 1.3.6.1.2.1.33.1.6.2.1.*(2).* using method: Normal

From device at: APC UPS using data hint: Std.ASN

Save in local objects starting at BI 2 saving up to this count: 24

Repeat this process every 15 seconds. Set optional object timeout to 20 seconds.

☒ Enable this table walk only when index object BI 26 is set to a value of 1

Table WalkRules Enabled: 3 Insert Delete

13.3 Table Walk Errors

If the table walk is unsuccessful, the table OID, device, and an error code are displayed here for the walk rule that ran into trouble.

Local Objects		BACnet		SNMP		System	
Local MIB		Client Setup		Client Data		Trap Sender	
Client Data		Errors: Read Maps		Errors: Write Maps		Errors: Table Walk	

<< Top Next >

Rule #	Table OID	Remote Device	Starting Object Name	Error Code
1	1.3.6.1.2.1.33.1.6.2.1.*(2).*	APC UPS	upsAlarmBatteryBad	207

Reset Errors

The following errors are possible. Except for error code 207, all of these error codes apply to any SNMP activity including client Get or Set activity defined by the SNMP

client read and write rules. Error 207 applies only to table walking, and means that the Normal method of table walking did not find enough OIDs in the table to satisfy the requested count of local objects to fill.

Standard SNMP error codes returned by remote Agent:

- 1 = SNMP_ERROR_tooBig
- 2 = SNMP_ERROR_noSuchName
- 3 = SNMP_ERROR_badValue
- 4 = SNMP_ERROR_readOnly
- 5 = SNMP_ERROR_genErr
- 6 = SNMP_ERROR_noAccess
- 7 = SNMP_ERROR_wrongType
- 8 = SNMP_ERROR_wrongLength
- 9 = SNMP_ERROR_wrongEncoding
- 10 = SNMP_ERROR_wrongValue
- 11 = SNMP_ERROR_noCreation
- 12 = SNMP_ERROR_inconsistentValue
- 13 = SNMP_ERROR_resourceUnavailable
- 14 = SNMP_ERROR_commitFailed
- 15 = SNMP_ERROR_undoFailed
- 16 = SNMP_ERROR_authorizationError
- 17 = SNMP_ERROR_notWritable
- 18 = SNMP_ERROR_inconsistentName

Client generated errors:

- 201 = No response from remote Agent (server)
- 202 = No such name (implicit)
- 203 = Unable to interpret application data
- 204 = Reply does not match request
- 205 = There is a problem with the rule configuration
- 206 = Unable to build or parse SNMP PDU
- 207 = Table walk came up short of expected count

If the error code indicates no response, the device status (SNMP Client Setup Devices page) will provide an additional indication of a connection related error.



14. Configuring SNMP Trap Receiver

14.1 Trap Receive Devices

The Devices page under Trap Receiver provides IP addresses of devices that will be sending traps to this Babel Buster. Traps will not be recognized by the trap rules if the IP address is not recognized in the list provided here.

The number of traps received from this device is counted. The counts reflect the number of traps received from this IP address since the last reset (or power up). Click Reset to reset the count. Both total traps and traps recognized by rules in the table are tabulated.

The screenshot shows the Babel Buster 3 web interface. The top banner is the same as the one above. Below it is a navigation bar with tabs: "Local Objects", "BACnet", "SNMP", "System", "Local MIB", "Client Setup", "Client Data", "Trap Sender", and "Trap Receiver". The "Trap Receiver" tab is selected. Under this tab, there are two sub-tabs: "Devices" and "Trap Rule". The "Devices" sub-tab is active. It shows a form for configuring a device. The "Source Device #" field contains the value "1". To the right of this field are "Update", "< Prev", and "Next >" buttons. Below this, the "IP Address" field contains "192.168.1.20". At the bottom, there are two fields: "Traps Received: 4" and "Recognized: 1". To the right of these fields is a "Reset" button.

14.2 Trap Receive Rules

Local Objects		BACnet		SNMP		System	
Local MIB		Client Setup		Client Data		Trap Sender	
Devices		Trap Rule					

Showing 1 to 2 of 2 Update < Prev Next >

Rule #	Dev #	v1 Trap #	Trap Variable Name (OID)	Data Hint	Result Object	Fixed Value	Timeout (Sec)	Timeout Value
1	1	6:1	1.3.6.1.2.1.33.2	Std. ASN	BI 1	<input checked="" type="checkbox"/> 1.000000	75	0.000000
2	0	0:0		Std. ASN	None	<input type="checkbox"/> 0.000000	0	0.000000

Rule # 1 Remove Insert Before

Rules for recognizing received traps are defined on this page. Rule number simply tells you where you're at on the list of Trap receiver rules. Click "next" and "prev" to scroll through the list. To advance directly to a specific rule, enter the desired number in the "Showing" box, then click Update.

Device number specifies which device on the SNMP Client Devices list this trap will be received from. If traps are received from devices not listed on the Devices page, they will be discarded. Setting device number to zero will result in the rule being removed from the list the next time the configuration file is saved.

If the trap received is a v1 (version 1) trap, you can specify the trap number expected as "a:b" where "a" is the generic trap number (for example, 6 for private enterprise) and "b" is the specific trap number as found in the device's MIB file. Leave this set to 0:0 if no v1 trap number is expected or should be disregarded. (Note: SNMP v2c traps will not have this number, and if trap type is detected as v2c, then this entry will be disregarded.)

The variable name that is expected to be found in the trap should be given as an OID for v2 traps, and optionally for v1 traps. If this OID is not found in the trap message, no further action is taken for this trap rule. If no OID is given, especially for v1 traps, then the trap number alone is enough to qualify this trap, and you should use the Fixed Value as the value to be placed into the result object.

Special case use of Trap Name OID: If the trap is v1 and trap numbers are given, then this OID, if provided, will be used to check the enterprise OID of the trap message received. If the OID is provided in the rule, then it must match the v1 trap enterprise OID before the rule will be considered successful.

Data hint helps the data parser figure out what the variable is. If ASN encoding is recognized as other than an octet string, the hint will be disregarded. If an octet string is found, then the parser needs to know if it should be treated as RFC 6340 floating point. If no hint is given (standard ASN), then the octet string will be treated as an ASCII character string, in which case ASCII to numeric conversion will be attempted automatically if the result object is numeric (string will simply be copied if result object is a character string type object). Note that standard, well-known ASN types are recognized as well as the NetSnmp ASN type for opaque float.

If a matching variable name is found, and the data parser succeeds in parsing the data

value, then it will be placed into the object identified as "Result Object". However, if Fixed Value is checked, then whatever value is entered next to the check box will be placed into the result object just because the variable name was found. In the case of v1 traps, if no OID is given, then the trap number alone is enough to cause the fixed value to be placed into the result object.

Timeout is optional, and will be in effect if a non-zero time in seconds is provided. If given, then the timeout value will be placed into the result object after this amount of time has elapsed since receiving the trap. If another of the same trap is received, the timer will be reset. Only after this time has expired without receiving the trap, but after receiving it at least once, the timeout value is placed into the result object.

Click Update when all entries have been made. After all configuration has been entered, be sure to go to the Config File page and click Save so that changes are retained through power cycles.

To insert or delete trap rules in the table, enter the trap number and click the appropriate button. To add a new rule to the end of the list when already at the end, simply click the Next button at the top.

14.3 Trap Receive Examples for SNMPv1

The trap receive rules illustrated in the following example will set Binary Input 1 to Active when the "on battery" trap is received from RFC 1628. This trap (per RFC 1628) will repeat every 60 seconds until power is restored. Therefore, we have set a timeout for just longer than 60 seconds so that our Binary Input object returns to Inactive after the alarm traps stop occurring.

Rule #	Dev #	v1 Trap #	Trap Variable Name (OID)	Data Hint	Result Object	Fixed Value	Timeout (Sec)	Timeout Value
1	1	6:1	1.3.6.1.2.1.33.2	Std. ASN	BI 1	<input checked="" type="checkbox"/> 1.000000	75	0.000000
2	0	0:0		Std. ASN	None	<input type="checkbox"/> 0.000000	0	0.000000

Showing 1 to 2 of 2

Update < Prev Next >

Rule # 1 Remove Insert Before

Upon receiving the trap, our Binary Input will become Active as illustrated here.

Local Objects		BACnet		SNMP		System	
Analog		Binary		Multi-State		Actions	
Input Objects		Output Objects		Value Objects			

Binary Input Objects Showing objects from

Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Text
<u>1</u>	upsOnBattery UPS on battery trap detected	N	Active	0	0,0,0,0	UPS is on battery

In addition to the trap number itself, the trap includes some varbinds (data pairs consisting of an OID and the data value identified by that OID). Rule #2 in the next example will capture the remaining minutes of battery power sent by the UPS with each of those trap messages sent every 60 seconds. Although this is still a v1 trap, it is important to leave the trap number set to 0:0 so that the varbind is processed by the rule rather than the trap number.

Local Objects		BACnet		SNMP		System	
Local MIB		Client Setup		Client Data		Trap Sender	
Devices		Trap Rule					

Showing to 3 of 3

Rule #	Dev #	v1 Trap #	Trap Variable Name (OID)	Data Hint	Result Object	Fixed Value	Timeout (Sec)	Timeout Value
1	<input type="text" value="1"/>	6:1	1.3.6.1.2.1.33.2	Std. ASN ▾	BI 1	<input checked="" type="checkbox"/> 1.000000	75	0.000000
2	<input type="text" value="1"/>	0:0	1.3.6.1.2.1.33.1.2.3	Std. ASN ▾	AI 15	<input type="checkbox"/> 0.000000	0	0.000000
3	<input type="text" value="0"/>	0:0		Std. ASN ▾	None	<input type="checkbox"/> 0.000000	0	0.000000

Rule #

Now each trap will set Binary Input 1 to Active, and also place the minutes remaining in Analog Input 15.

An APC UPS includes both RFC 1628 and their own private enterprise MIB which starts at 1.3.6.1.4.1.318. The APC MIB sends enterprise specific trap number 5 when the UPS switches to battery power, and trap number 9 when the UPS switches off battery power (utility power restored). In the example below, rules 2 and 3 will set Binary Input 27 to Active when the "on battery" trap is received, and clear Binary Input 27 to Inactive when the "not on battery" trap is received.

Devices

Trap Rule

Showing 1 to 4 of 4

Update

< Prev

Next >

Rule #	Dev #	v1 Trap #	Trap Variable Name (OID)	Data Hint	Result Object	Fixed Value	Timeout (Sec)	Timeout Value
1	1	6:1	1.3.6.1.2.1.33.2	Std. ASN	BI 1	<input checked="" type="checkbox"/> 1.000000	75	0.000000
2	1	6:5		Std. ASN	BI 27	<input checked="" type="checkbox"/> 1.000000	0	0.000000
3	1	6:9		Std. ASN	BI 27	<input checked="" type="checkbox"/> 0.000000	0	0.000000
4	0	0:0		Std. ASN	None	<input type="checkbox"/> 0.000000	0	0.000000

Rule # 1

Remove

Insert Before

Binary Input 27 has been defined as follows.

Local Objects	BACnet	SNMP	System
Analog	Binary	Multi-State	Actions
Input Objects	Output Objects	Value Objects	

Binary Input # 27 Update < Prev Next >

Reliability: 0 Status: 0,0,0,0 Device Link: --- Out of Service: ☐ Deconfigure: ☐

Object name upsOnBatteryAPC Force ☐ Present Value Inactive

Description APC MIB UPS on battery trap

Active Text: UPS on battery Inctive Text: UPS not on battery

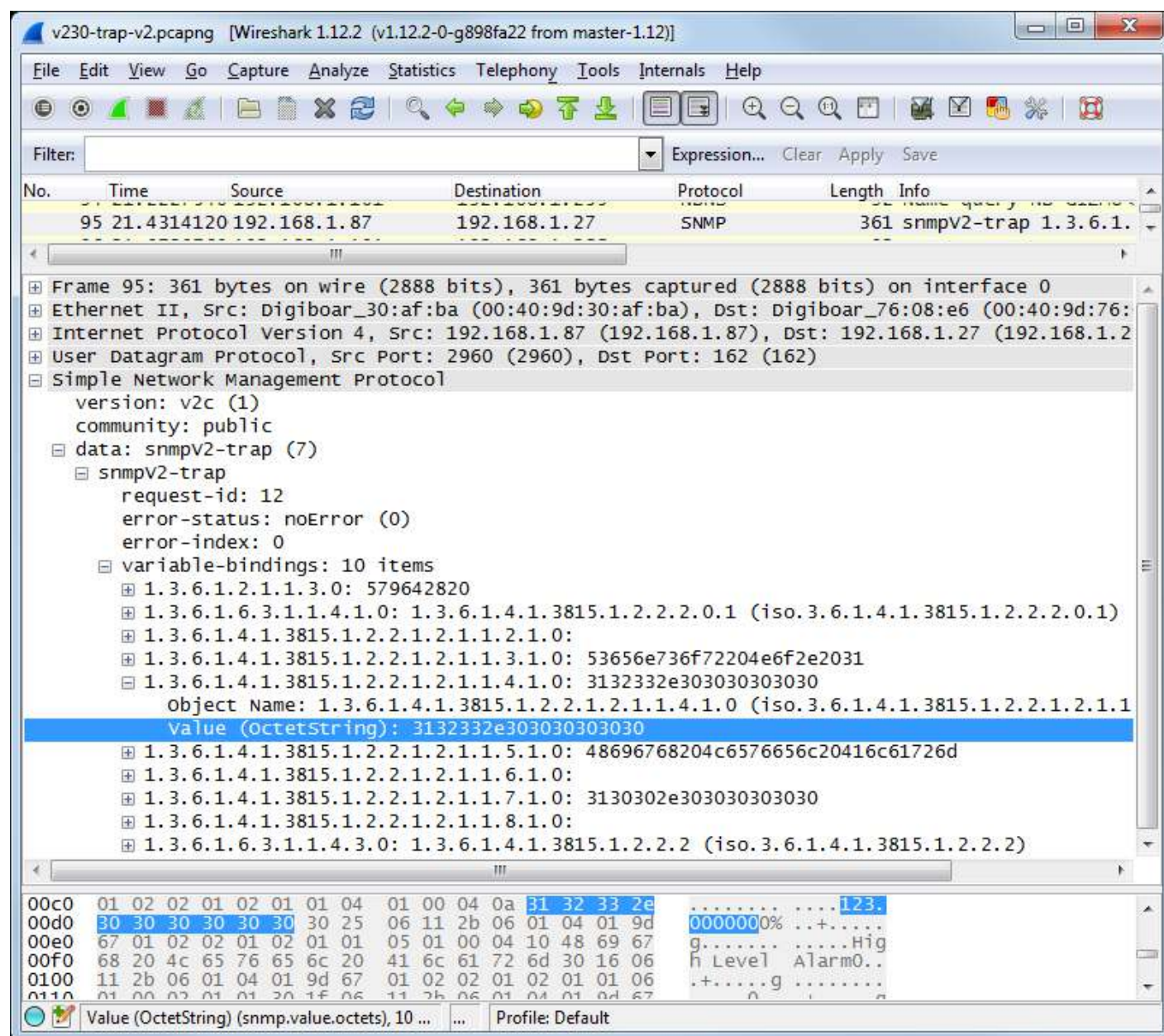
14.4 Trap Receive Example for SNMPv2

Most of what is discussed for SNMPv1 trap receive rules also applies to SNMPv2. The only real difference is that SNMPv2 does not send trap numbers, therefore you cannot use the v1 trap "a:b" number for v2 traps. Anything that will be recognized as a v2 trap will require a Trap Variable Name (OID). Other than trap number, everything else about configuring the v2 trap receive rule works the same as for v1.

Local Objects		BACnet		SNMP		System									
Local MIB		Client Setup		Client Data		Trap Sender		Trap Receiver							
Devices		Trap Rule													
Showing 1 to 3 of 3										Update		< Prev		Next >	
Rule #	Dev #	v1 Trap #	Trap Variable Name (OID)		Data Hint	Result Object	Fixed Value	Timeout (Sec)	Timeout Value						
1	1	0:0	1.3.6.1.4.1.3815.1.2.2.1.2.1.1.4.1		Std. ASN ▾	AV 1	<input type="checkbox"/> 0.000000	0	0.000000						
2	0	0:0			Std. ASN ▾	None	<input type="checkbox"/> 0.000000	0	0.000000						
3	0	0:0			Std. ASN ▾	None	<input type="checkbox"/> 0.000000	0	0.000000						
Rule # 1										Remove		Insert Before			

The Wireshark capture of a trap from this v2 device is illustrated below. In this case, the OID in the trap rule identifies a numeric (floating piont) value that is sent as an

ASCII string.



Numeric conversion of the Octet String is attempted automatically since the data hint is "Std. ASN" (standard ASN). The result is the correct value placed in a floating point Analog Value object.

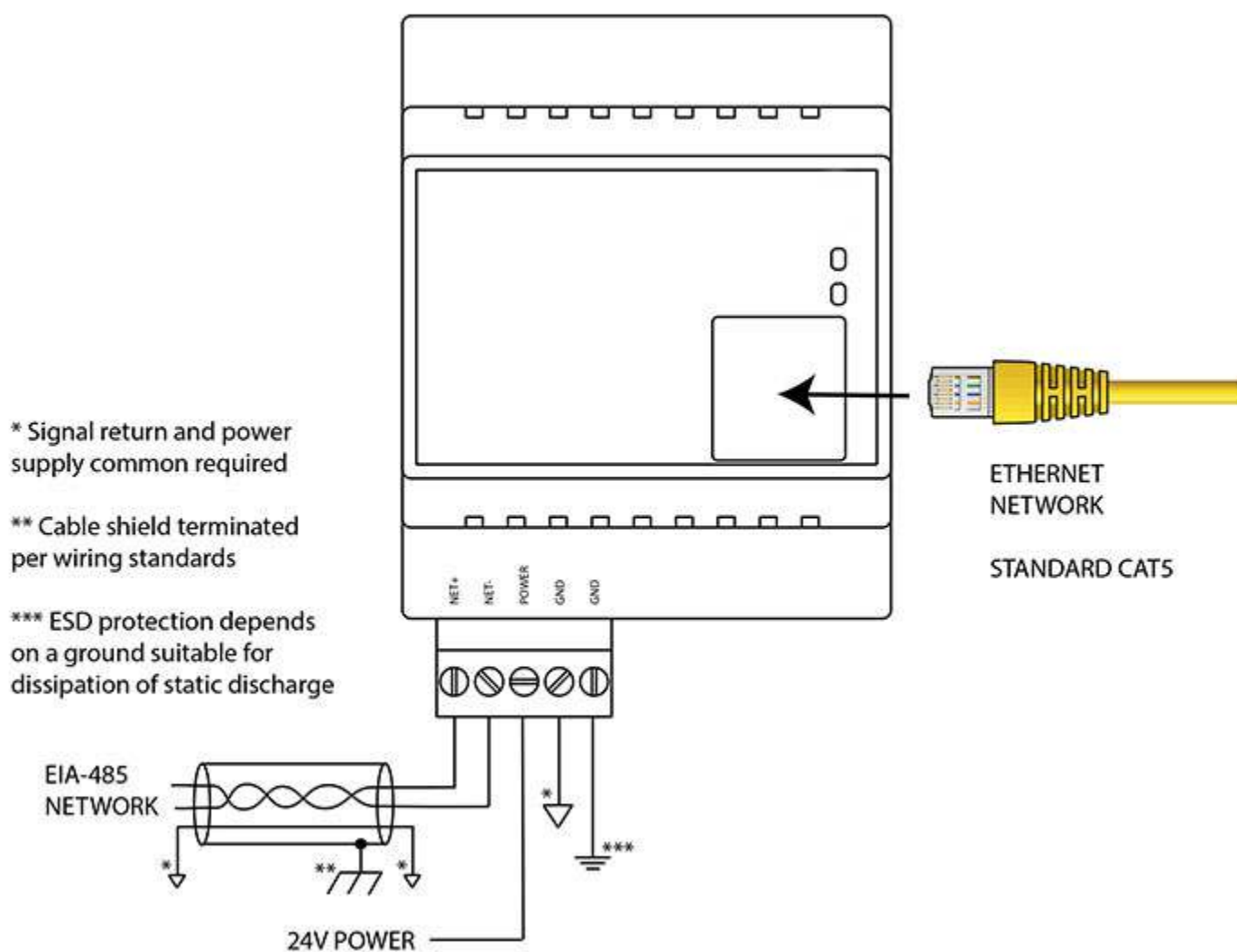
Local Objects		BACnet		SNMP		System	
Analog		Binary		Multi-State		Actions	
Input Objects		Output Objects		Value Objects			
Analog Value Objects		Showing objects from 1		Update		< Prev Next >	
Object	Object Name Object Description	Out of Service	Present Value	Reliability	Status	Units	
1	Analog Value 1	N	123.0000	0	0,0,0,0	no_units	
2	Analog Value 2	N	0.000000	0	0,0,0,0	no_units	



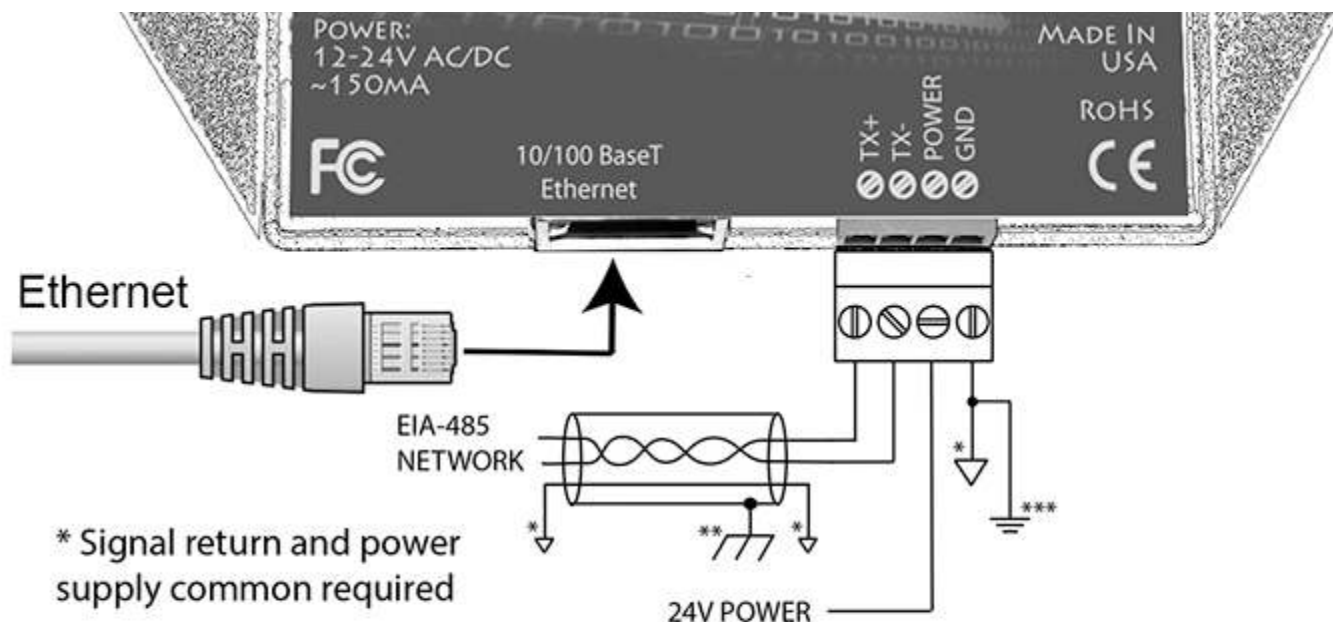
Appendix A Hardware Details

A.1 Wiring

Wiring for the Babel Buster BB3-7302 is illustrated below.



Wiring for the MX-73 is illustrated below.



* Signal return and power supply common required

** Cable shield terminated per wiring standards

*** ESD protection depends on a ground suitable for dissipation of static discharge

Wire the gateway as illustrated. Follow all conventional standards for wiring of EIA-485 networks when connecting the MS/TP EIA-485 (RS485) network. This includes use and termination of shield, termination of the network, and grounding.

IMPORTANT: Although EIA-485 (RS485) is thought of as a 2-wire network, you **MUST** include a third conductor connected to GND or common at each device so that all devices are operating at close to the same ground potential. Proper grounding of equipment should ensure proper operation without the third conductor; however, proper grounding often cannot be relied upon. If large common mode voltages are present, you may even need to insert optically isolated repeaters between EIA-485 devices.

Use standard CAT5 cables for Ethernet connections. Use control wire as applicable for local electrical codes for connecting the 24V (AC or DC) power supply.

Note that in addition to connecting power supply common to a GND terminal, you must also connect a GND terminal to earth ground in order to ensure proper ESD protection.

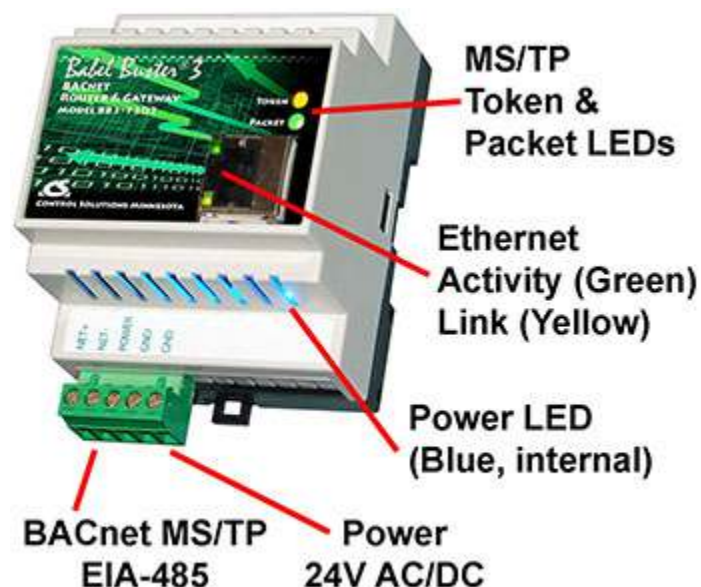
A.2 Front Panel LED Indicators

A.2.1 BB3-7302 LED Indicators

Power-up LED behavior: On power up, the Packet LED will remain on solid red for

about 20 seconds, then the Token and Packet LEDs will do a "lamp test" where Token is yellow and Packet is Red simultaneously for about 1 second, and then both Token and Packet turn green simultaneously for about 1 second. The LEDs will then begin to operate according to their normal functionality.

If the gateway is powered up with no Ethernet network attached, the Token and Packet LEDs will continue to blink yellow/red indefinitely until the network is connected. Then they will flash green once, and then proceed to normal operation.



Babel Buster BB3-7302 Token and Packet LEDs reflect BACnet MS/TP traffic, and the Ethernet activity LED will indicate network traffic in general.

Babel Buster BB3-7302 LEDs indicate as follows (LEDs are bi-color):

Token	Flashes green each time the MS/TP token is passed. Flashes yellow each time the gateway sends a Poll For Master on the MS/TP link. (If it appears to be solid yellow, this is an indication that it is never passing the token, meaning it is not seeing any other devices on the network to pass the token to.)
Packet	Flashes green each time a request is sent on the MS/TP link, or a reply is received on the MS/TP link. Flashes red if the request resulted in an error.
Ethernet Activity	Green LED is on solid during portions of the boot-up process, and then flashes briefly when Ethernet network traffic is detected.
Ethernet Link	Yellow LED indicates an Ethernet link is present. This indicator will light if a link is present regardless of processor or network activity. If not lit, check network wiring.

Power	Blue LED (internal) on any time power is present and internal power supply is functioning.
-------	--

A.2.2 MX-73 LED Indicators

Power-up LED behavior: On power up, the Token, Packet and Error LEDs will remain off for about 20 seconds, then all three LEDs will do a "lamp test" where they all turn on simultaneously for about 1 second. The LEDs will then begin to operate according to their normal functionality.



Babel Buster MX-73 Token, Packet and Error LEDs reflect BACnet MS/TP traffic, and the Ethernet activity LED will indicate network traffic in general.

Babel Buster MX-73 LEDs indicate as follows (LEDs are each a single color):

Error (red)	Flashes red upon receipt of an error indication in a reply message, or upon recognition of an error in response to a request that was received. If the BACnet client is configured to query other MS/TP devices, then the error LED can also indicate a timeout waiting for a reply.
Packet (yellow)	Flashes yellow each time a request is sent or received. Also flashes each time a "poll for master" is sent. If the MS/TP network is not connected, the yellow LED will appear as if on solid because all it is doing is polling for masters in an attempt to find something else to talk to.
Token (green)	Flashes each time the token is passed. Frequent but intermittent flashes of green indicate normal network traffic on the MS/TP link.
Ethernet Activity	Green LED is on solid during portions of the boot-up process, and then flashes briefly when Ethernet network traffic is

	detected.
Ethernet Link	Yellow LED indicates an Ethernet link is present. This indicator will light if a link is present regardless of processor or network activity. If not lit, check network wiring.
Status	Blue LED (internal) on any time power is present and internal power supply is functioning.

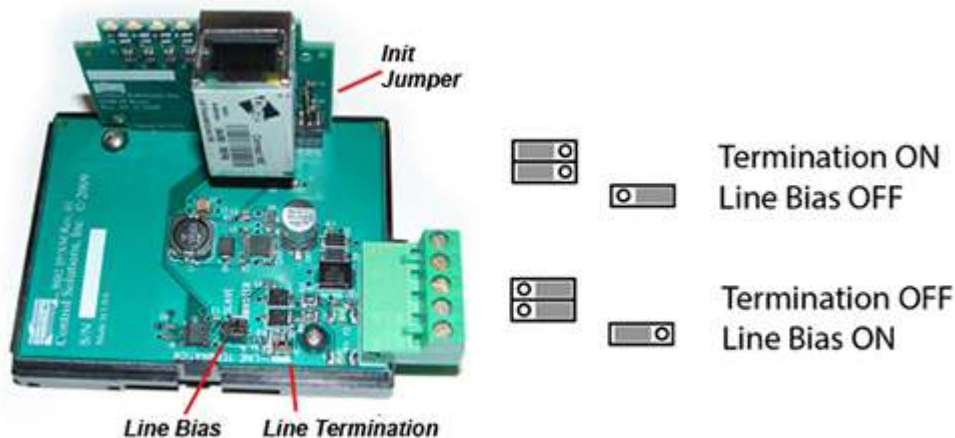
A.3 RS-485 Line Termination & Bias

Enable line termination only when this device is placed at the end of the network. Termination should only be enabled at two points on the network, and these two points must be specifically the end points.

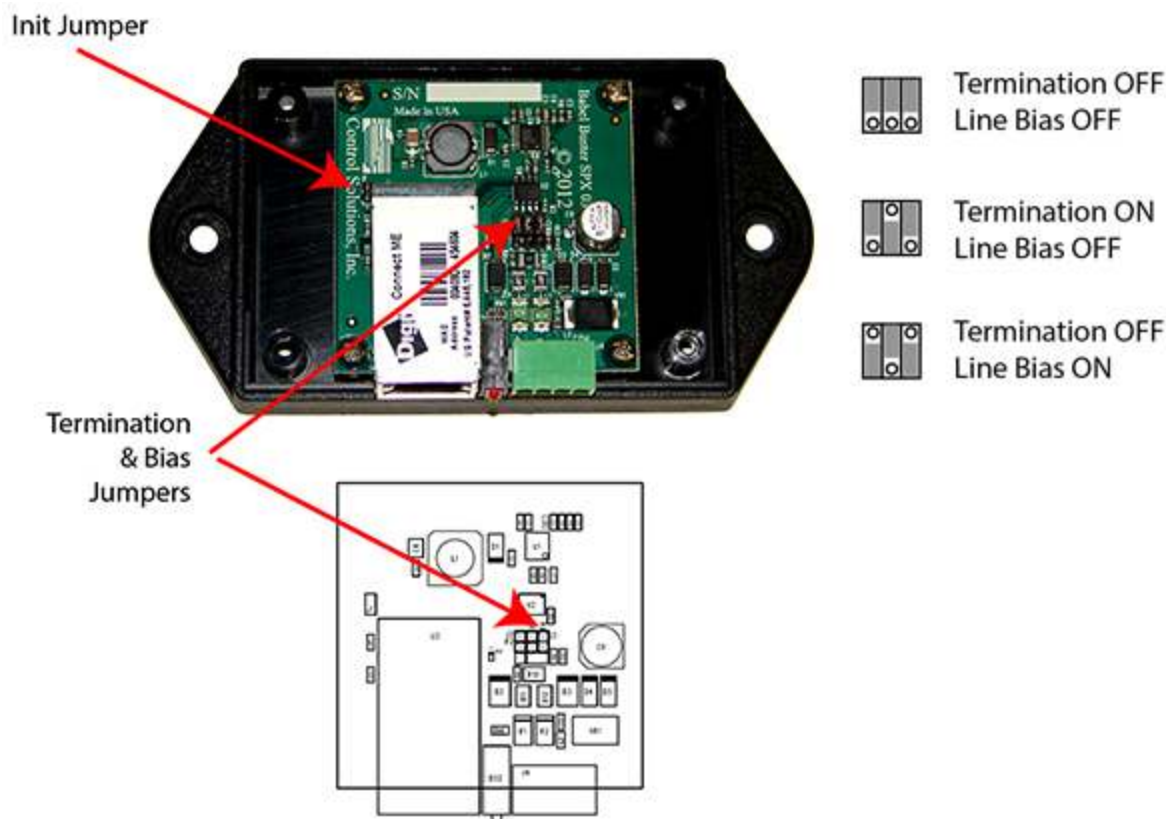
Enable line bias when needed. Line bias should only be enabled at one point on the network, and does not have to be the end point. Line bias holds the line in a known neutral state when no devices are transmitting. Without bias, the transition from offline to online by a transmitter can look like a false start bit and cause loss of communication.

The line conditioning options are enabled when the respective shunt is moved to the position indicated by the diagrams below.

Jumper locations for Babel Buster BB3-7302:



Jumper locations for Babel Buster MX-73:



A.4 Soft Configuration Reset

Soft reset should be used to remove all configuration information any time you do have the ability to connect to the gateway's web user interface. The "Clear Configuration" action is described in Section 3.1.5. Using the forced hard reset should only be used as a last resort if you are unable to connect to the gateway because the SSL certificates are invalid for a secure connection or you are unable to recover the lost IP address.

A.5 Discovering Lost IP Address

You can use Wireshark to discover a lost IP address if the gateway is still functional. Connect the gateway directly to your PC running Wireshark using a cross-over cable (or standard CAT5 cable if your PC supports auto-MDIX). With Wireshark running, power up the gateway.

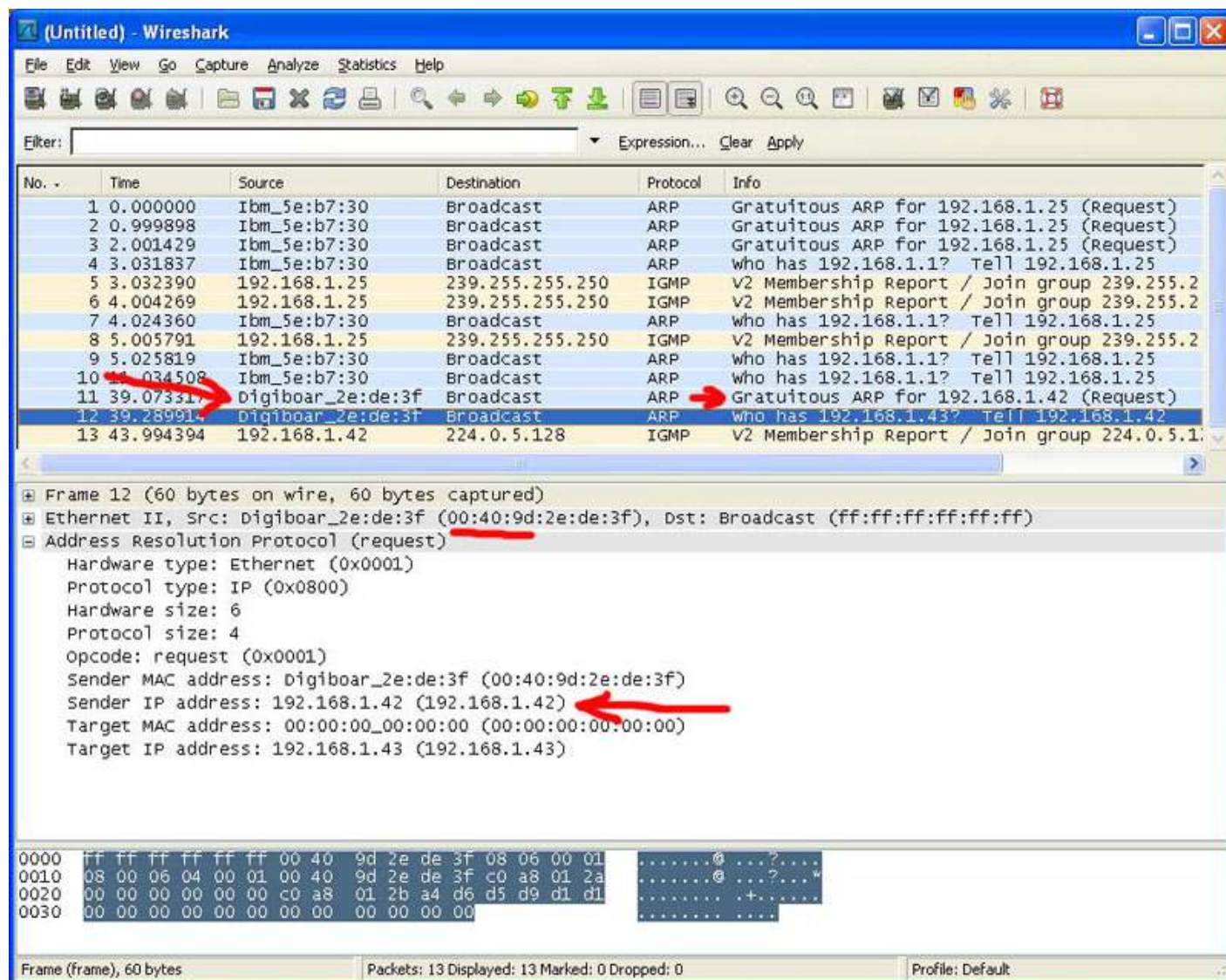
Upon power up, BB3-7302/MX-73 will ping its own IP address one or more times. This is part of its duplicate address resolution mechanism. If it finds another device with its own IP address, it will set its own IP address to a default pseudo-random address generally starting with 192.

Wait until you are certain BB3-7302/MX-73 has booted up, or wait 2-3 minutes to be sure if you don't recognize the bootup LED sequence. Now look for the ARP packets and note what IP address they came from. This is your device. (To make sure it is your device, connect only the BB3-7302/MX-73 to your PC while doing this exercise.)

Your device will have a MAC address that starts with 00:40:9D, also labeled with a

source that starts with "Digiboar_". This label comes from the fact that the server modules used on Control Solutions IP products are made by Digi International, previously known as "Digiboard".

There will usually be one or more "pings" or ARP packets to the device's own IP address, and one last ping to its own address plus one. In the illustration here, the BB3-7302 is located at 192.168.1.42.



A.6 Forced Hard Configuration Reset

IMPORTANT: Before considering the forced hard reset, be sure you have considered soft configuration reset, or discovering lost IP address if applicable.

The "Init" jumper inside the BB3-7302/MX-73 serves two purposes, and what it does depends on whether you apply the jumper before or after the BB3-7302/MX-73 boots up.

Hard Configuration Reset:

Installing the jumper after bootup causes the BB3-7302/MX-73 to do a hard reset on

its configuration memory. The IPv4 address will be reset to 10.0.0.101. The root password will be reset to the original default password. After clearing all configuration, the BB3-7302/MX-73 will automatically restart. Remove the jumper when you see the indication of restart after about 30 seconds, which is both LEDs coming on solid on the RJ45 Ethernet connector and remaining on for a couple of seconds. If you miss the start of reboot, both LEDs on the RJ45 will come on and stay on. It will now be attempting the firmware update, but you can abort that by simply powering down the BB3-7302/MX-73. If both LEDs on the RJ45 jack come on and remain on, remove the jumper and then power cycle the BB3-7302/MX-73.

Once you have regained access to the device, go to the File Manager page, execute the Clear All configuration action, then select the file named as "Boot configuration" and execute the Save XML Config File action to wipe out any configuration normally saved in the XML configuration file.

Note: The forced hard reset will restore HTTP web access and disable HTTPS web access. The forced hard reset will also restore FTP access to allow FTP firmware uploads if needed.

Note: The hard reset of configuration also means all of your resource allocations are reset to original factory defaults. If you want resource allocations that are different, you will need to repeat the allocation setup as described in Section 3.4.

Firmware Update Recovery:

Installing this jumper prior to power-up causes the server to go into TFTP firmware update mode. Normally you would perform a firmware update by simply uploading a new image.bin file (provided by Control Solutions tech support) using the BB3-7302/MX-73's internal FTP server and a command line FTP session on your PC (Linux or Windows command line). Detailed instructions are included in the zip file that also contains the applicable image.bin file.

Should the FTP upload fail for some reason, then you need to resort to the TFTP upload method as the fallback method. Full details on how to go about this can be found under the topic "Restoring a corrupt application image" at <https://info.csimn.com>.

Additional maintenance page:

Go to [http\(s\)://10.0.0.101/html/pgRestoreAddr.html](http(s)://10.0.0.101/html/pgRestoreAddr.html) to find the following page (substituting your IP address). It serves two purposes as noted below, which ideally you will never have a use for.



File System Wipe:

On rare occasion, the Flash file system has been observed to get corrupted as a result of losing power while a write operation was in progress. This is most effectively confirmed by opening a command prompt FTP session (Windows 10 PowerShell) to try to view the files in the Flash file system. If FTP fails to show any files, in addition to other problems saving or loading files, it may be that the file system has gotten corrupted. If this happens, go to the page pictured above, and enter the Reformat key, then click Wipe, and then power cycle the device (or restart from the File Manager page). The reformat key is 55AAAA55. Simply type that into the window next to the Wipe button.

MAC Address Restore:

In the event the MAC address has been reset due to NVRAM checksum failure, this page will permit restoring the MAC address to its original address as printed on the component label internal to this device, or on the default password label found on the outside or on external documentation included with the device.

If the MAC address is deemed to be valid, the window will be labeled "Valid MAC Address" and you will not be allowed to change it. If the MAC address is deemed to be invalid, the window will be labeled "Restore MAC Address" and you should then enter the correct MAC address and click Restore. A restart is then needed.

A.7 Firmware Update Notes

The most up to date firmware is shipped with all new devices. This isn't like a new laptop where you spent the first half a day updating software on a computer you thought was brand new. If you believe you have discovered an issue that you believe a firmware update might fix, contact technical support first to confirm whether that is the case, and then to get a login to the firmware update support site.

The brute force approach to updating firmware using TFTP as noted in the section above is always available, but the more graceful approach is to use FTP to upload the new image.bin file. There is one minor problem: The upload wants to buffer the entire file in RAM while it proceeds to reprogram the Flash memory. **If the memory**

utilization indicated on the Resources page in your device is above about 30%, the FTP upload will fail, and thus the firmware update will not take place.

You have two choices: (1) Use the TFTP approach, or (2) Temporarily reconfigure your gateway to use a minimum of resources to free up space to temporarily buffer the image.bin file upload.

More detailed instructions for the FTP upload are included in the zip file you will download to obtain the firmware update. Instructions for the TFTP upload are available in our knowledgebase at <https://info.csimn.com>.



Appendix B Trouble Shooting

B.1 File Upload Errors

If you get a "File upload error: -1" message, click the browser's "back" button, then simply click the View button to view any file (does not matter which file), and then click browser's "back" button again to return to the File Manager page. This gets the browser and HTTP server back in sync, and this requirement generally happens only once following power-up.

If you get a different persistent file upload error, check the space available versus the size of the file you are trying to upload. Available file space is displayed on the File Manager page as "Free space". The free space indicated is approximate. If close to zero, try deleting some files first.

B.2 BACnet Trouble Shooting

B.2.1 General Trouble Shooting

Refer to Appendix A for additional details about hardware.



(a) Does the device have power? A blue LED inside the case, visible through the air vent slots, will be on if there is power present.

(b) Is the Ethernet Link LED (yellow) on? If not, check your network cable. If LED activity on the RJ45 jack for the Ethernet connection is normal, then refer to Trouble Shooting BACnet IP below.

(c) Is the Token LED flashing? If not, or you are not sure of its behavior, refer to the Token LED discussion that follows under Trouble Shooting MS/TP.

B.2.2 Error Codes, Reliability Codes

Attempts to communicate with other devices by gateway client functions will result in error codes being set as non-zero reliability codes in the affected object. Additional error information can be viewed on the respective client errors pages in the Web UI.

When the reliability code is non-zero, the fault flag is also set. Therefore, the Status Flags indication will typically be "F,T,F,F" any time the reliability code is something other than zero (zero means no errors to report).

Specific error responses will generally be due to incorrect configuration. However, timeouts or "no response" errors can be more difficult to trouble shoot as these can be a combination of configuration errors in the local device, configuration errors on the network in general, or wiring problems.

B.2.3 Auto-Reset Errors

Reliability codes will "latch" by default and require that you read the Reliability property in order to reset it to zero, assuming the problem has gone away. Once the non-zero reliability code has been read (by reading the Reliability property), it will reset to zero the next time the object is updated, provided the problem has been resolved.

Since many systems do not automatically read Reliability codes, but do automatically respond to the Fault Status Flag associated with the non-zero reliability code, an auto-reset option is available. This option is indicated as "Allow fault self-reset without Ack" on the BACnet Settings page. When selected, reliability codes will return to zero as soon as the problem has been resolved, regardless of whether the non-zero reliability code was ever acknowledged by reading it.

Many systems will report an object as "offline" because its fault status bit is set. It is not actually offline, and is in fact communicating just fine trying to tell you that there is a problem. But many front end systems don't recognize this and blindly claim "offline" as a result of the fault bit in the status property. If you are having this issue as the result of communication errors related to polling other devices using the gateway's client maps, try setting the Auto-Reset Errors option here.

B.2.4 Trouble Shooting BACnet IP

BACnet IP is typically easier to get running than MS/TP just because Ethernet is pretty straight forward. The most frequent problem is "no response" or timeout. The most common cause of this problem for BACnet IP is a network configuration problem, such

as incorrect IP address or IP address that cannot be reached as configured. The problem sometimes lies outside the Babel Buster and may require consulting with the IT personnel responsible for the network if on a large network.

The subnet mask determines what part of the IP address constitutes the domain, and all devices on the same network must be on the same domain before they can communicate.

Obviously two devices being assigned the same IP address is going to cause trouble. If you can communicate at all with a device having a duplicate IP address, it will be intermittent, and potentially erratic as the other device having the same IP address may be responding to your queries.

If you are connecting via one or more routers, then everything that applies to routing issues will apply to your device. A complete discussion of NAT routing, BACnet routing, etc, is beyond the scope of this document - you should refer these questions to your IT administrator when applicable.

Once the BB3-7301 is communicating BACnet IP, then next area for possible concern is with the BACnet client. If the gateway is supposed to be polling other IP devices, but the data does not appear correct, the first thing to check is the reliability code. Any reliability code other than zero is a problem. Refer to the list at the bottom of any of the Data Objects pages for explanation of the non-zero codes. If the reliability code indicates that an error was returned by the server (meaning the other BACnet device you are trying to query), then refer to the BACnet Diagnostics page for additional error information.

B.2.5 Trouble Shooting BACnet MS/TP

The Token LED can give you some clues about MS/TP network connectivity. The Token LED will flash yellow each time it sends a Poll For Master, and green each time it passes the token.

Token LED remains OFF:

The MS/TP port does not get initialized until the configuration file is successfully loaded. If a faulty XML file was loaded, then the port might not get initialized and the Token LED will remain off. To confirm that the configuration file is good, go to the File Manager page, select your boot configuration file, select Load XML Config File, and Execute. If you get no error messages, then the file is good and the port should be initialized.

When an MS/TP device first comes online, it is only listening - waiting for its MAC address to be polled in a Poll For Master message. If port settings like baud rate are wrong and the port is constantly hearing only noise as a result, then the Token LED will never flash because this MS/TP device is never given a chance to talk. Another less likely possibility is that this MS/TP device's MAC address has been set outside the Max Master range of other devices already on the network. Hearing nothing but noise can also be caused by RS-485 wiring problems. For a detailed discussion of RS-485 wiring, refer to this link: https://csimn.com/CSI_pages/RS-485-FAQ.html.

Token LED solid ON Yellow:

The local MS/TP device will first listen to see if it hears anything on the network. If not, then it will begin polling for masters looking for somebody to talk to. If no other devices ever reply, and the only thing going on is more Poll For Master messages, then the Token LED will appear to be on solid yellow because it is only flashing yellow very rapidly.

Token LED flashing yellow and green:

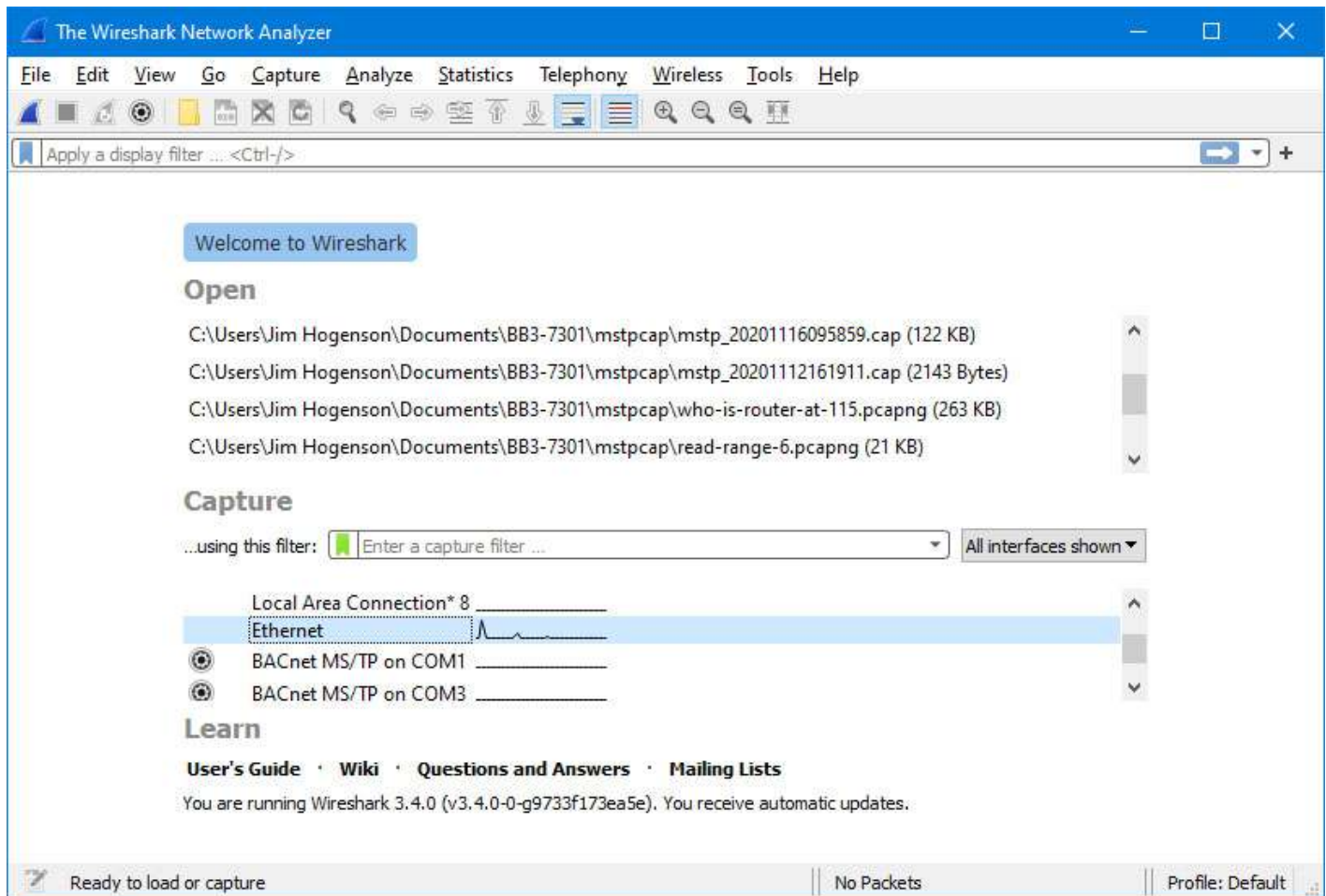
The normal behavior will be continuous flashing green at some rate dependent on overall network load along with occasional flashes of yellow when it polls for other masters. If and how often you see flashes of yellow depend on MAC address distribution across the network.

Token LED flashing yellow and green with long pauses OFF:

This is an indication of almost working but there is confusion somewhere on the network either due to misconfiguration, or due to noise resulting from RS-485 wiring issues. One misconfiguration that can cause this appearance is an incorrect Max Master setting in one or more devices on the network. The Max Master setting **MUST** be identical in all devices. In addition, the MAC address **MUST** be unique in all devices, and duplicate MAC addresses can also cause this behavior. If you are certain all configuration is correct and are still seeing this behavior, then you need to start looking at wiring. For a detailed discussion of RS-485 wiring, refer to this link: https://csimn.com/CSI_pages/RS-485-FAQ.html.

B.2.6 Using Wireshark

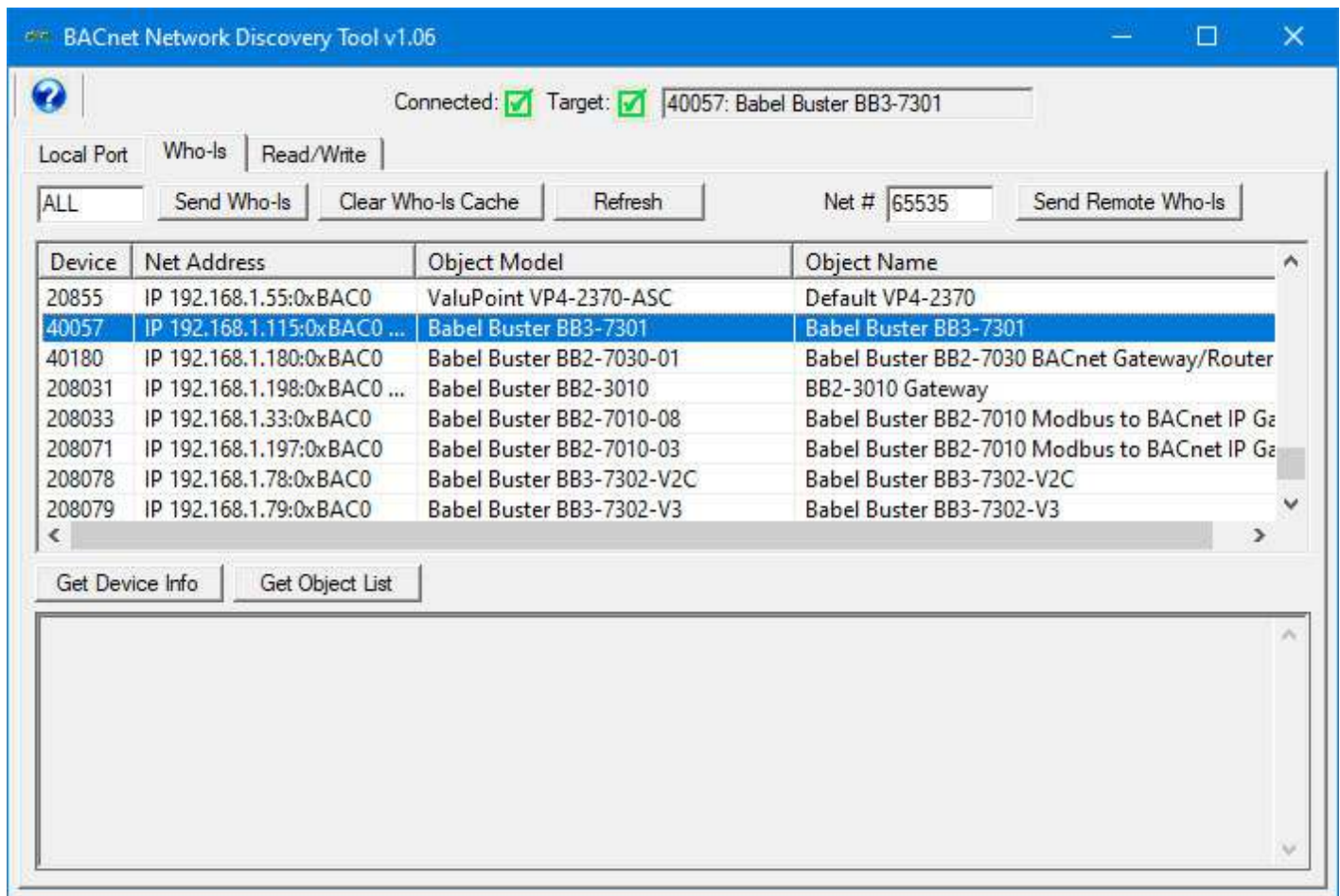
One of the most useful tools for diagnosing BACnet IP problems is Wireshark. You can get a free copy at www.wireshark.org. Additional important information about Wireshark can be found in **Appendix F** of this user guide. When you start Wireshark, the startup screen appears as follows (as of this writing). Click on Ethernet (for BACnet IP) or BACnet MS/TP on COMx to begin capturing traffic.



B.2.7 Using Network Discovery Tool

Control Solutions has created a Network Discovery Tool to perform simple diagnostics on BACnet devices and networks. It works with BACnet IP using your PC's Ethernet connection - assuming your PC is connected to the BACnet IP network. It will also work with MS/TP using the Control Solutions MTX002 USB to MS/TP adapter.

The Network Discovery Tool can be used to discover devices on the network.



The Network Discovery Tool can be used to read and write properties in any BACnet device on the network.

BACnet Network Discovery Tool v1.06

Connected: ☒ Target: ☒ 40057: Babel Buster BB3-7301

Local Port | Who-Is | Read/Write

Object Type: Analog Value

Object Instance: 1

Property: Present Value 0

Array Index: (leave blank for no index)

Data Type: Real

Priority: None ☐ Relinquish

Write Data: 12.34

12.340000

Buttons: Read Property, Write Property, Send Raw APDU

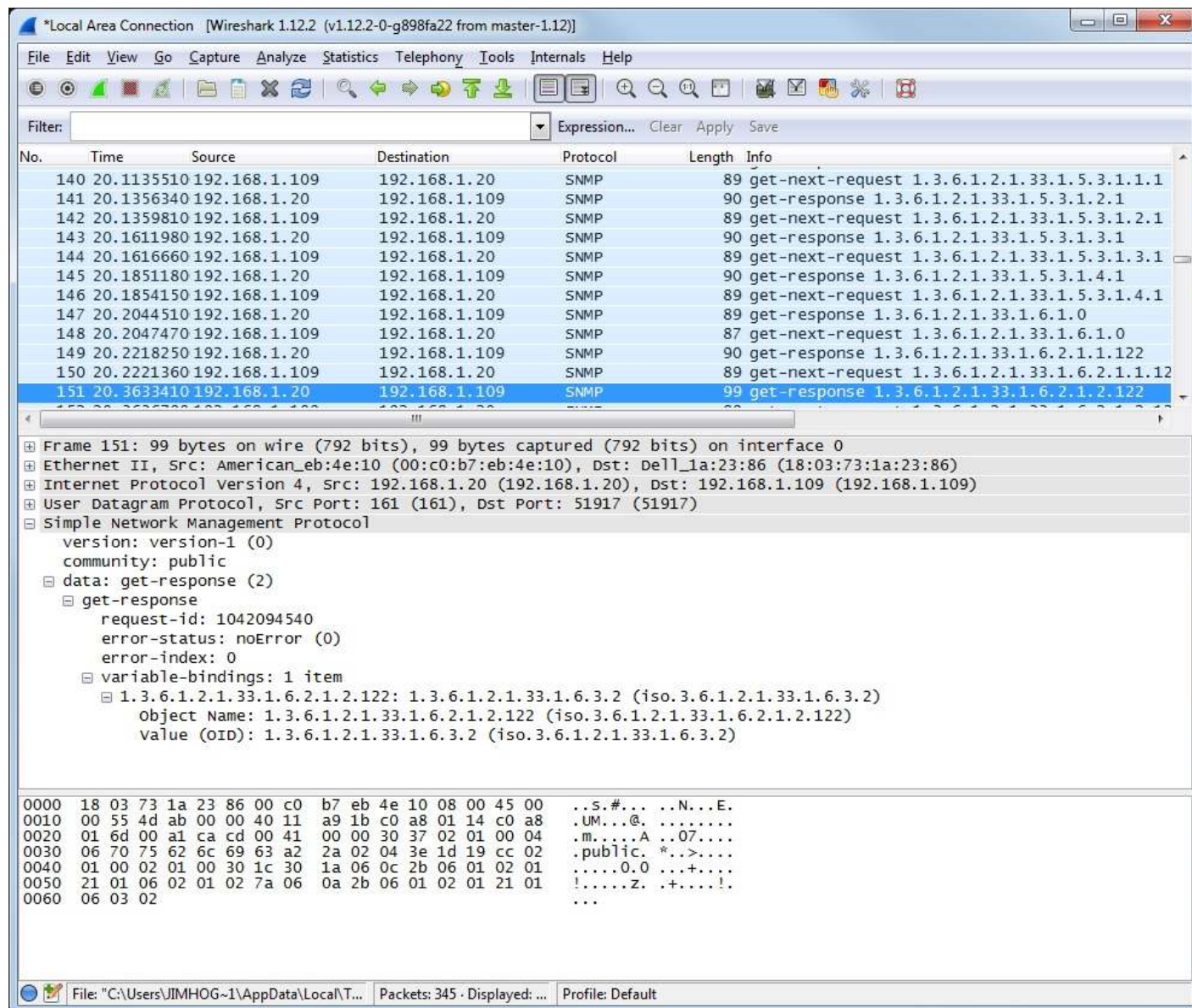
The complete user guide for the Network Discover Tool can be viewed by clicking the blue "?" help icon in the upper left corner of the tool. You can obtain a free copy of the tool on the Tool Links page under Support at the Control Solutions web site (https://csimn.com/CSI_pages/Tools.html).

B.3 SNMP Trouble Shooting

Assuming you have IP addresses configured correctly and the SNMP ports are open through any routers and firewalls between devices, the most common cause of not communicating is a mismatching community string for SNMPv2 or incorrect user credentials for SNMPv3. Without the correct credentials, most devices will simply ignore the request, making it look as if there is no connection when in fact there is nothing wrong with the connection.

Another common oversight is that when adding local registers to the local MIB, you need to click the Reload SNMP button at the bottom of the Local MIB pages to cause SNMP to reload its internal tables with the new configuration you just entered. The Local MIB pages are effectively a list of instructions for loading the SNMP MIB, but the MIB is not automatically rebuilt every time you add another line to the Local MIB pages. To reload the MIB according to the list of instructions you provided on the Local MIB web pages, you need to click Reload SNMP. (Note: The Reload SNMP button is found on multiple pages, but they all perform the same function and any of the Reload SNMP buttons will reload all branches of the MIB.)

Wireshark can be a useful tool for analyzing SNMP issues. Refer to Appendix F for more about Wireshark. The screen shot below illustrates SNMP on the network.





Appendix C CSV File Formats

HINT: If you get "table full" errors while importing CSV files, you might not have sufficient resources allocated. You may need to increase some counts on the Resources page.

C.1 BACnet IP Client Read/Write Maps

The CSV file for configuring BACnet IP client read and write maps should contain a single header line with the labels indicated below, and content as applicable.

Header Line Label	Notes	Description of Use
RW	-	Enter 'R' to Read from a remote device, or 'W' to Write to a remote device.
REMOTEOBJTYPE	-	Indicate the type of local BACnet object (see C.2) that should be read or written at the remote device. In addition to the object types recognized as local objects, the client may read remote Accumulator objects referenced as type "AC".
REMOTEOBJNUM	-	Indicate the remote object number that goes along with object type in the previous column.
PROPERTY	-	Specify by BACnet code the object property (see Appendix E) that should be read. The most common is Present Value, whose code is 85.
INDEX	-	If the property to be read/written is an array, then an array index is needed. Specify "no index" by entering zero in the CSV column. Otherwise enter 1 or greater, and note that actual index values will be offset by -1 when applied by the BACnet client.
DEVNUM	-	Specify the device number where the remote object is to be found. This number is used to look up a device in the BACnet Client Device table which contains the device's BACnet Device Instance, or static binding if applicable, etc.
SCALE	-	Data is multiplied by this scale factor after read from a remote device or before being written to a remote device.
OFFSET	-	This offset is added to the data value after read from a remote device or before being written to a remote device.

POLL	-	Specify a periodic poll time in seconds (fractions of sections are recognized).
OBJTYPE	-	Indicate the type of local BACnet object (see C.2) where data read from a remote device will be placed, or where data written to a remote device will be taken from.
OBJNUM	-	Indicate the object number that goes along with object type in the previous column.
DEFAULT	1	When READING: The default value will be stored into the local object after the given number of read failures if the fail count (MAXFAIL) is non-zero.
MAXFAIL	1	If non-zero, sets the maximum number of times that a read attempt may fail before the default value will be placed in the local object. Setting the count to zero will disable the default, and the object will retain the most recent value obtained.
DATATYPE	2	Provide the data type code that the remote object being written expects to receive. 1=Boolean, 2=Unsigned Integer, 3=Signed Integer, 4=Real, 9=Enumerated (note that 5, 6, 7, 8 are not used here)
PRIORITY	2	If writing to a commandable object, then a priority (1-16) must be provided.
MAXQUIET	2	If using 'send on delta', to guarantee that the remote device will be written at least occasionally even if the data does not change, enter a maximum quiet time (in seconds).
MINQUIET	2	If using 'send on delta', and the delta increment is small, the result can be a large amount of network traffic. To limit network traffic, provide a MINQUIET time (in seconds) that must elapse between transmission of changed values.
DELTA	2	The local object/register data may be written to the remote device periodically, or when the local value changes, or both. To send upon change (send on delta), provide a DELTA value as the amount by which the local object must change before being written to the remote device. Leave blank if send on delta should not be used.

Notes:

- 1) Applies only to Read maps (enter zero as place holder for Write maps)
- 2) These apply only to Write maps (enter zero as place holder for Read maps)

The minimum required header line for BACnet IP must include RW, REMOTEOBJTYPE, REMOTEOBJNUM, PROPERTY, DEVNUM, OBJTYPE, OBJNUM. All other columns are optional.

A typical CSV configuration file as viewed in Excel would look something like this:

	A	B	C	D	E	F	G	H	I	J	K	L
1	RW	REMOTEO	REMOTEO	PROPERTY	INDEX	DEVNUM	SCALE	OFFSET	POLL	OBJTYPE	OBJNUM	
2	R	AI	1	0	0	1	0	0	30	AI	1	
3	R	AI	2	0	0	1	0	0	30	AI	2	
4	R	AI	3	0	0	1	0	0	30	AI	3	
5	R	AI	4	0	0	1	0	0	30	AI	4	
6	R	AI	5	0	0	1	0	0	30	AI	5	
7	R	AI	6	0	0	1	0	0	30	AI	6	
8	R	AI	7	0	0	1	0	0	30	AI	7	
9	R	AI	8	0	0	1	0	0	30	AI	8	
10	R	AI	9	0	0	1	0	0	30	AI	9	
11	R	AI	10	0	0	1	0	0	30	AI	10	

The same CSV file as viewed with a simple text editor like Notepad would look like this:

```

bip-maps.csv - Notepad
File Edit Format View Help
RW,REMOTE,OBJTYPE,REMOTE,OBJNUM,PROPERTY,INDEX,DEVNUM,SCALE,OFFSET,POLL,OBJTYPE,OBJNUM
R,AI,1,0,0,1,0,0,30,AI,1
R,AI,2,0,0,1,0,0,30,AI,2
R,AI,3,0,0,1,0,0,30,AI,3
R,AI,4,0,0,1,0,0,30,AI,4
R,AI,5,0,0,1,0,0,30,AI,5
R,AI,6,0,0,1,0,0,30,AI,6
R,AI,7,0,0,1,0,0,30,AI,7
R,AI,8,0,0,1,0,0,30,AI,8
R,AI,9,0,0,1,0,0,30,AI,9
R,AI,10,0,0,1,0,0,30,AI,10
R,AI,11,0,0,1,0,0,30,AI,11
R,AI,12,0,0,1,0,0,30,AI,12
R,AI,13,0,0,1,0,0,30,AI,13
R,AI,14,0,0,1,0,0,30,AI,14
R,AI,15,0,0,1,0,0,30,AI,15
R,AI,16,0,0,1,0,0,30,AI,16
R,AI,1,0,0,2,0,0,30,AI,17
R,AI,2,0,0,2,0,0,30,AI,18
R,AI,3,0,0,2,0,0,30,AI,19
R,AI,4,0,0,2,0,0,30,AI,20
R,AI,5,0,0,2,0,0,30,AI,21
R,AI,6,0,0,2,0,0,30,AI,22
R,AI,7,0,0,2,0,0,30,AI,23
R,AI,8,0,0,2,0,0,30,AI,24
Ln 1, Col 1 100% Windows (CRLF) UTF-8

```

C.2 BACnet Object Types

CSV Label	BACnet Object Type
AI	Analog Input
AO	Analog Output
AV	Analog Value
BI	Binary Input
BO	Binary Output
BV	Binary Value
MI	Multistate Input
MO	Multistate Output
MV	Multistate Value



Appendix D BACnet Object Properties

D.1 Data Object Properties (Analog, Binary, Multi-state)

The following properties are found in the Analog, Binary, and Multi-state types of Input, Output, and Value objects. Some properties apply only to certain object types as noted where applicable.

Property	Encoding
Object_Identifier (75)	BACnetObjectIdentifier
Object_Name (77) (W)	CharacterString "Analog Input <i>n</i> "
Object_Type (79)	BACnetObjectType ENUMERATED: analog-input (0) analog-output (1) analog-value (2) binary-input (3) binary-output (4) binary-value (5) device (8) multi-state-input (13) multi-state-output (14) multi-state-value (19)
Present_Value (85) (W)	REAL (analog objects) ENUMERATED (binary objects) Unsigned (multi-state objects) (no index) (priority required when writing commandable objects) (input objects writeable only when out of service)
Status_Flags (111)	BACnetStatusFlags BIT STRING: fault(1), out-of-service(3)
Event_State (36)	BACnetEventState ENUMERATED: normal(0), fault(1)
Reliability (103)	BACnetReliability ENUMERATED: normal(0)

	<i>Vendor specific:</i> BACnet IP client, device timeout (82) BACnet IP client, error returned by server (83) SNMP client, no response from agent (91) SNMP client, agent returned 'no such name' error (92) SNMP client, unable to parse data returned (93) SNMP client, reply did not match request (94) SNMP client, invalid table walk configuration (95) SNMP client, unable to parse PDU returned (96) SNMP client, table walk came up short on data (97) SNMP client, agent returned error other than 'no such name' (98)
Description (28) (W)	CharacterString
Out_Of_Service (81) (W)	BOOLEAN
COV_Increment (22) (W)	REAL (analog objects only)
Priority_Array (87)	BACnetPriorityArray (commandable objects only) SEQUENCE SIZE (16) OF BACnetPriorityValue REAL (each element, analog output objects) ENUMERATED (each element, binary output objects) Unsigned (each element, multi-state output objects)
Relinquish_Default (104) (W)	REAL (analog objects) ENUMERATED (binary objects) Unsigned (multi-state objects)
Polarity (84)	BACnetPolarity (binary objects only) ENUMERATED: normal(0)
Number_Of_States (74)	Unsigned (multi-state objects only)
Units (117)	BACnetEngineeringUnits (analog objects only)
Active_Text (4) (W)	CharacterString (binary objects only)
Inactive_Text (46) (W)	CharacterString (binary objects only)
State_Text (110) (W)	BACnetARRAY[N] of CharacterString (multi-state objects only)

D.2 Device Object Properties

The following properties are found in the Device object.

Property	Encoding
Object_Identifier (75)	BACnetObjectIdentifier

Object_Name (77)	CharacterString
Object_Type (79)	BACnetObjectType ENUMERATED: device (8)
System_Status (112)	BACnetDeviceStatus
Vendor_Name (121)	CharacterString
Vendor_Identifier (120)	Unsigned16 (should always return 208)
Model_Name (70)	CharacterString
Firmware_Revision (44)	CharacterString
Application_Software_Version (12)	CharacterString
Protocol_Version (98)	Unsigned
Protocol_Revision (139)	Unsigned
Protocol_Services_Supported (97)	BACnetServicesSupported
Protocol_Object_Types_Supported (96)	BACnetObjectTypesSupported
Object_List (76)	BACnetARRAY[N] of BACnetObjectIdentifier
Max_APDU_Length_Accepted (62)	Unsigned
Segmentation_Supported (107)	BACnetSegmentation
APDU_Timeout (11)	Unsigned
Number_Of_APDU_Retries (73)	Unsigned
Device_Address_Binding (30)	BACnetLIST of BACnetAddressBinding
Database_Revision (155)	Unsigned
Description (28)	CharacterString
Location (58)	CharacterString
Max_Segments_Accepted (167)	Unsigned
APDU_Segment_Timeout (10)	Unsigned
Active_COV_Subscriptions (152)	BACnetLIST of BACnetCOVSubscription
Last_Restart_Reason (196)	BACnetRestartReason
Restart_Notification_Recipients (202)	BACnetLIST of BACnetRecipient
Time_Of_Device_Restart (203)	BACnetTimeStamp

Max_Master (64)	Unsigned (0..127)
Max_Info_Frames (63)	Unsigned
Auto_Slave_Discovery (169)	BACnetARRAY[N] of BOOLEAN
Manual_Slave_Address_Binding (170)	BACnetLIST of BACnetAddressBinding
Slave_Address_Binding (171)	BACnetLIST of BACnetAddressBinding
Slave_Proxy_Enable (172)	BACnetARRAY[N] of BOOLEAN



Appendix E BACnet Codes

E.1 BACnet Object Property Codes

BACnet property type codes may be found in your copy of the BACnet protocol specification, ANSI/ASHRAE Standard 135. That document is copyrighted, but the C enumeration shown below for reference is taken from open source code available under GPL at <http://sourceforge.net>, and provides essentially the same information (copyrighted by Steve Karg, licensed under GPL as noted at <http://sourceforge.net>).

```
typedef enum {
    PROP_ACKED_TRANSITIONS = 0,
    PROP_ACK_REQUIRED = 1,
    PROP_ACTION = 2,
    PROP_ACTION_TEXT = 3,
    PROP_ACTIVE_TEXT = 4,
    PROP_ACTIVE_VT_SESSIONS = 5,
    PROP_ALARM_VALUE = 6,
    PROP_ALARM_VALUES = 7,
    PROP_ALL = 8,
    PROP_ALL_WRITES_SUCCESSFUL = 9,
    PROP_APDU_SEGMENT_TIMEOUT = 10,
    PROP_APDU_TIMEOUT = 11,
    PROP_APPLICATION_SOFTWARE_VERSION = 12,
    PROP_ARCHIVE = 13,
    PROP_BIAS = 14,
    PROP_CHANGE_OF_STATE_COUNT = 15,
    PROP_CHANGE_OF_STATE_TIME = 16,
    PROP_NOTIFICATION_CLASS = 17,
    PROP_BLANK_1 = 18,
    PROP_CONTROLLED_VARIABLE_REFERENCE = 19,
    PROP_CONTROLLED_VARIABLE_UNITS = 20,
    PROP_CONTROLLED_VARIABLE_VALUE = 21,
    PROP_COV_INCREMENT = 22,
    PROP_DATE_LIST = 23,
    PROP_DAYLIGHT_SAVINGS_STATUS = 24,
    PROP_DEADBAND = 25,
    PROP_DERIVATIVE_CONSTANT = 26,
    PROP_DERIVATIVE_CONSTANT_UNITS = 27,
    PROP_DESCRIPTION = 28,
    PROP_DESCRIPTION_OF_HALT = 29,
    PROP_DEVICE_ADDRESS_BINDING = 30,
    PROP_DEVICE_TYPE = 31,
```

```
PROP_EFFECTIVE_PERIOD = 32,  
PROP_ELAPSED_ACTIVE_TIME = 33,  
PROP_ERROR_LIMIT = 34,  
PROP_EVENT_ENABLE = 35,  
PROP_EVENT_STATE = 36,  
PROP_EVENT_TYPE = 37,  
PROP_EXCEPTION_SCHEDULE = 38,  
PROP_FAULT_VALUES = 39,  
PROP_FEEDBACK_VALUE = 40,  
PROP_FILE_ACCESS_METHOD = 41,  
PROP_FILE_SIZE = 42,  
PROP_FILE_TYPE = 43,  
PROP_FIRMWARE_REVISION = 44,  
PROP_HIGH_LIMIT = 45,  
PROP_INACTIVE_TEXT = 46,  
PROP_IN_PROCESS = 47,  
PROP_INSTANCE_OF = 48,  
PROP_INTEGRAL_CONSTANT = 49,  
PROP_INTEGRAL_CONSTANT_UNITS = 50,  
PROP_ISSUE_CONFIRMED_NOTIFICATIONS = 51,  
PROP_LIMIT_ENABLE = 52,  
PROP_LIST_OF_GROUP_MEMBERS = 53,  
PROP_LIST_OF_OBJECT_PROPERTY_REFERENCES = 54,  
PROP_LIST_OF_SESSION_KEYS = 55,  
PROP_LOCAL_DATE = 56,  
PROP_LOCAL_TIME = 57,  
PROP_LOCATION = 58,  
PROP_LOW_LIMIT = 59,  
PROP_MANIPULATED_VARIABLE_REFERENCE = 60,  
PROP_MAXIMUM_OUTPUT = 61,  
PROP_MAX_APDU_LENGTH_ACCEPTED = 62,  
PROP_MAX_INFO_FRAMES = 63,  
PROP_MAX_MASTER = 64,  
PROP_MAX_PRES_VALUE = 65,  
PROP_MINIMUM_OFF_TIME = 66,  
PROP_MINIMUM_ON_TIME = 67,  
PROP_MINIMUM_OUTPUT = 68,  
PROP_MIN_PRES_VALUE = 69,  
PROP_MODEL_NAME = 70,  
PROP_MODIFICATION_DATE = 71,  
PROP_NOTIFY_TYPE = 72,  
PROP_NUMBER_OF_APDU_RETRIES = 73,  
PROP_NUMBER_OF_STATES = 74,  
PROP_OBJECT_IDENTIFIER = 75,  
PROP_OBJECT_LIST = 76,  
PROP_OBJECT_NAME = 77,  
PROP_OBJECT_PROPERTY_REFERENCE = 78,  
PROP_OBJECT_TYPE = 79,  
PROP_OPTIONAL = 80,  
PROP_OUT_OF_SERVICE = 81,  
PROP_OUTPUT_UNITS = 82,  
PROP_EVENT_PARAMETERS = 83,  
PROP_POLARITY = 84,
```



```
PROP_PRESENT_VALUE = 85,  
PROP_PRIORITY = 86,  
PROP_PRIORITY_ARRAY = 87,  
PROP_PRIORITY_FOR_WRITING = 88,  
PROP_PROCESS_IDENTIFIER = 89,  
PROP_PROGRAM_CHANGE = 90,  
PROP_PROGRAM_LOCATION = 91,  
PROP_PROGRAM_STATE = 92,  
PROP_PROPORTIONAL_CONSTANT = 93,  
PROP_PROPORTIONAL_CONSTANT_UNITS = 94,  
PROP_PROTOCOL_CONFORMANCE_CLASS = 95,          /* deleted in version 1  
revision 2 */  
PROP_PROTOCOL_OBJECT_TYPES_SUPPORTED = 96,  
PROP_PROTOCOL_SERVICES_SUPPORTED = 97,  
PROP_PROTOCOL_VERSION = 98,  
PROP_READ_ONLY = 99,  
PROP_REASON_FOR_HALT = 100,  
PROP_RECIPIENT = 101,  
PROP_RECIPIENT_LIST = 102,  
PROP_RELIABILITY = 103,  
PROP_RELINQUISH_DEFAULT = 104,  
PROP_REQUIRED = 105,  
PROP_RESOLUTION = 106,  
PROP_SEGMENTATION_SUPPORTED = 107,  
PROP_SETPOINT = 108,  
PROP_SETPOINT_REFERENCE = 109,  
PROP_STATE_TEXT = 110,  
PROP_STATUS_FLAGS = 111,  
PROP_SYSTEM_STATUS = 112,  
PROP_TIME_DELAY = 113,  
PROP_TIME_OF_ACTIVE_TIME_RESET = 114,  
PROP_TIME_OF_STATE_COUNT_RESET = 115,  
PROP_TIME_SYNCHRONIZATION_RECIPIENTS = 116,  
PROP_UNITS = 117,  
PROP_UPDATE_INTERVAL = 118,  
PROP_UTC_OFFSET = 119,  
PROP_VENDOR_IDENTIFIER = 120,  
PROP_VENDOR_NAME = 121,  
PROP_VT_CLASSES_SUPPORTED = 122,  
PROP_WEEKLY_SCHEDULE = 123,  
PROP_ATTEMPTED_SAMPLES = 124,  
PROP_AVERAGE_VALUE = 125,  
PROP_BUFFER_SIZE = 126,  
PROP_CLIENT_COV_INCREMENT = 127,  
PROP_COV_RESUBSCRIPTION_INTERVAL = 128,  
PROP_CURRENT_NOTIFY_TIME = 129,  
PROP_EVENT_TIME_STAMPS = 130,  
PROP_LOG_BUFFER = 131,  
PROP_LOG_DEVICE_OBJECT = 132,  
/* The enable property is renamed from log-enable in  
   Addendum b to ANSI/ASHRAE 135-2004(135b-2) */  
PROP_ENABLE = 133,  
PROP_LOG_INTERVAL = 134,
```

```
PROP_MAXIMUM_VALUE = 135,  
PROP_MINIMUM_VALUE = 136,  
PROP_NOTIFICATION_THRESHOLD = 137,  
PROP_PREVIOUS_NOTIFY_TIME = 138,  
PROP_PROTOCOL_REVISION = 139,  
PROP_RECORDS_SINCE_NOTIFICATION = 140,  
PROP_RECORD_COUNT = 141,  
PROP_START_TIME = 142,  
PROP_STOP_TIME = 143,  
PROP_STOP_WHEN_FULL = 144,  
PROP_TOTAL_RECORD_COUNT = 145,  
PROP_VALID_SAMPLES = 146,  
PROP_WINDOW_INTERVAL = 147,  
PROP_WINDOW_SAMPLES = 148,  
PROP_MAXIMUM_VALUE_TIMESTAMP = 149,  
PROP_MINIMUM_VALUE_TIMESTAMP = 150,  
PROP_VARIANCE_VALUE = 151,  
PROP_ACTIVE_COV_SUBSCRIPTIONS = 152,  
PROP_BACKUP_FAILURE_TIMEOUT = 153,  
PROP_CONFIGURATION_FILES = 154,  
PROP_DATABASE_REVISION = 155,  
PROP_DIRECT_READING = 156,  
PROP_LAST_RESTORE_TIME = 157,  
PROP_MAINTENANCE_REQUIRED = 158,  
PROP_MEMBER_OF = 159,  
PROP_MODE = 160,  
PROP_OPERATION_EXPECTED = 161,  
PROP_SETTING = 162,  
PROP_SILENCED = 163,  
PROP_TRACKING_VALUE = 164,  
PROP_ZONE_MEMBERS = 165,  
PROP_LIFE_SAFETY_ALARM_VALUES = 166,  
PROP_MAX_SEGMENTS_ACCEPTED = 167,  
PROP_PROFILE_NAME = 168,  
PROP_AUTO_SLAVE_DISCOVERY = 169,  
PROP_MANUAL_SLAVE_ADDRESS_BINDING = 170,  
PROP_SLAVE_ADDRESS_BINDING = 171,  
PROP_SLAVE_PROXY_ENABLE = 172,  
PROP_LAST_NOTIFY_TIME = 173,  
PROP_SCHEDULE_DEFAULT = 174,  
PROP_ACCEPTED_MODES = 175,  
PROP_ADJUST_VALUE = 176,  
PROP_COUNT = 177,  
PROP_COUNT_BEFORE_CHANGE = 178,  
PROP_COUNT_CHANGE_TIME = 179,  
PROP_COV_PERIOD = 180,  
PROP_INPUT_REFERENCE = 181,  
PROP_LIMIT_MONITORING_INTERVAL = 182,  
PROP_LOGGING_DEVICE = 183,  
PROP_LOGGING_RECORD = 184,  
PROP_PRESCALE = 185,  
PROP_PULSE_RATE = 186,  
PROP_SCALE = 187,
```

```
PROP_SCALE_FACTOR = 188,  
PROP_UPDATE_TIME = 189,  
PROP_VALUE_BEFORE_CHANGE = 190,  
PROP_VALUE_SET = 191,  
PROP_VALUE_CHANGE_TIME = 192,  
/* enumerations 193-206 are new */  
PROP_ALIGN_INTERVALS = 193,  
PROP_GROUP_MEMBER_NAMES = 194,  
PROP_INTERVAL_OFFSET = 195,  
PROP_LAST_RESTART_REASON = 196,  
PROP_LOGGING_TYPE = 197,  
PROP_MEMBER_STATUS_FLAGS = 198,  
PROP_NOTIFICATION_PERIOD = 199,  
PROP_PREVIOUS_NOTIFY_RECORD = 200,  
PROP_REQUESTED_UPDATE_INTERVAL = 201,  
PROP_RESTART_NOTIFICATION_RECIPIENTS = 202,  
PROP_TIME_OF_DEVICE_RESTART = 203,  
PROP_TIME_SYNCHRONIZATION_INTERVAL = 204,  
PROP_TRIGGER = 205,  
PROP.UTC_TIME_SYNCHRONIZATION_RECIPIENTS = 206,  
/* enumerations 207-211 are used in Addendum d to ANSI/ASHRAE  
135-2004 */  
PROP_NODE_SUBTYPE = 207,  
PROP_NODE_TYPE = 208,  
PROP_STRUCTURED_OBJECT_LIST = 209,  
PROP_SUBORDINATE_ANNOTATIONS = 210,  
PROP_SUBORDINATE_LIST = 211,  
/* enumerations 212-225 are used in Addendum e to ANSI/ASHRAE  
135-2004 */  
PROP_ACTUAL_SHED_LEVEL = 212,  
PROP_DUTY_WINDOW = 213,  
PROP_EXPECTED_SHED_LEVEL = 214,  
PROP_FULL_DUTY_BASELINE = 215,  
/* enumerations 216-217 are used in Addendum i to ANSI/ASHRAE  
135-2004 */  
PROP_BLINK_PRIORITY_THRESHOLD = 216,  
PROP_BLINK_TIME = 217,  
/* enumerations 212-225 are used in Addendum e to ANSI/ASHRAE  
135-2004 */  
PROP_REQUESTED_SHED_LEVEL = 218,  
PROP_SHED_DURATION = 219,  
PROP_SHED_LEVEL_DESCRIPTIONS = 220,  
PROP_SHED_LEVELS = 221,  
PROP_STATE_DESCRIPTION = 222,  
/* enumerations 223-225 are used in Addendum i to ANSI/ASHRAE  
135-2004 */  
PROP_FADE_TIME = 223,  
PROP_LIGHTING_COMMAND = 224,  
PROP_LIGHTING_COMMAND_PRIORITY = 225,  
/* enumerations 226-235 are used in Addendum f to ANSI/ASHRAE  
135-2004 */  
PROP_DOOR_ALARM_STATE = 226,  
PROP_DOOR_EXTENDED_PULSE_TIME = 227,
```

```
PROP_DOOR_MEMBERS = 228,  
PROP_DOOR_OPEN_TOO_LONG_TIME = 229,  
PROP_DOOR_PULSE_TIME = 230,  
PROP_DOOR_STATUS = 231,  
PROP_DOOR_UNLOCK_DELAY_TIME = 232,  
PROP_LOCK_STATUS = 233,  
PROP_MASKED_ALARM_VALUES = 234,  
PROP_SECURED_STATUS = 235,  
/* enumerations 236-243 are used in Addendum i to ANSI/ASHRAE  
135-2004 */  
PROP_OFF_DELAY = 236,  
PROP_ON_DELAY = 237,  
PROP_POWER = 238,  
PROP_POWER_ON_VALUE = 239,  
PROP_PROGRESS_VALUE = 240,  
PROP_RAMP_RATE = 241,  
PROP_STEP_INCREMENT = 242,  
PROP_SYSTEM_FAILURE_VALUE = 243,  
/* enumerations 244-311 are used in Addendum j to ANSI/ASHRAE  
135-2004 */  
PROP_ABSENTEE_LIMIT = 244,  
PROP_ACCESS_ALARM_EVENTS = 245,  
PROP_ACCESS_DOORS = 246,  
PROP_ACCESS_EVENT = 247,  
PROP_ACCESS_EVENT_AUTHENTICATION_FACTOR = 248,  
PROP_ACCESS_EVENT_CREDENTIAL = 249,  
PROP_ACCESS_EVENT_TIME = 250,  
PROP_ACCESS_RULES = 251,  
PROP_ACCESS_RULES_ENABLE = 252,  
PROP_ACCESS_TRANSACTION_EVENTS = 253,  
PROP_ACCOMPANIED = 254,  
PROP_ACTIVATION_TIME = 255,  
PROP_ACTIVE_AUTHENTICATION_POLICY = 256,  
PROP_ASSIGNED_ACCESS_RIGHTS = 257,  
PROP_AUTHENTICATION_FACTOR_INPUT_LIST = 258,  
PROP_AUTHENTICATION_FACTORS = 259,  
PROP_AUTHENTICATION_POLICY_LIST = 260,  
PROP_AUTHENTICATION_POLICY_NAMES = 261,  
PROP_AUTHORIZATION_MODE = 262,  
PROP_BELONGS_TO = 263,  
PROP_CREDENTIAL_DISABLE = 264,  
PROP_CREDENTIAL_STATUS = 265,  
PROP_CREDENTIALS = 266,  
PROP_CREDENTIALS_IN_ZONE = 267,  
PROP_DAYS_REMAINING = 268,  
PROP_ENTRY_POINTS = 269,  
PROP_EXIT_POINTS = 270,  
PROP_EXPIRY_TIME = 271,  
PROP_EXTENDED_TIME_ENABLE = 272,  
PROP_FAILED_ATTEMPT_EVENTS = 273,  
PROP_FAILED_ATTEMPTS = 274,  
PROP_FAILED_ATTEMPTS_TIME = 275,  
PROP_FORMAT_CLASS_SUPPORTED = 276,
```

```
PROP_FORMAT_TYPE = 277,  
PROP_LAST_ACCESS_EVENT = 278,  
PROP_LAST_ACCESS_POINT = 279,  
PROP_LAST_CREDENTIAL_ADDED = 280,  
PROP_LAST_CREDENTIAL_ADDED_TIME = 281,  
PROP_LAST_CREDENTIAL_REMOVED = 282,  
PROP_LAST_CREDENTIAL_REMOVED_TIME = 283,  
PROP_LAST_USE_TIME = 284,  
PROP_LOCKDOWN = 285,  
PROP_LOCKDOWN_RELINQUISH_TIME = 286,  
PROP_MASTER_EXEMPTION = 287,  
PROP_MAX_FAILED_ATTEMPTS = 288,  
PROP_MEMBERS = 289,  
PROP_MASTER_POINT = 290,  
PROP_NUMBER_OF_AUTHENTICATION_POLICIES = 291,  
PROP_OCCUPANCY_COUNT = 293,  
PROP_OCCUPANCY_COUNT_ENABLE = 294,  
PROP_OCCUPANCY_COUNT_EXEMPTION = 295,  
PROP_OCCUPANCY_LOWER_THRESHOLD = 296,  
PROP_OCCUPANCY_LOWER_THRESHOLD_ENFORCED = 297,  
PROP_OCCUPANCY_STATE = 298,  
PROP_OCCUPANCY_UPPER_LIMIT = 299,  
PROP_OCCUPANCY_UPPER_LIMIT_ENFORCED = 300,  
PROP_PASSBACK_EXEMPTION = 301,  
PROP_PASSBACK_MODE = 302,  
PROP_PASSBACK_TIMEOUT = 303,  
PROP_POSITIVE_ACCESS_RULES = 304,  
PROP_READ_STATUS = 305,  
PROP_REASON_FOR_DISABLE = 306,  
PROP_THREAT_AUTHORITY = 307,  
PROP_THREAT_LEVEL = 308,  
PROP_TRACE_FLAG = 309,  
PROP_TRANSACTION_NOTIFICATION_CLASS = 310,  
PROP_USER_EXTERNAL_IDENTIFIER = 311,  
/* enumerations 312-313 are used in Addendum k to ANSI/ASHRAE  
135-2004 */  
PROP_CHARACTER_SET = 312,  
PROP_STRICT_CHARACTER_MODE = 313,  
/* enumerations 312-313 are used in Addendum k to ANSI/ASHRAE  
135-2004 */  
PROP_BACKUP_AND_RESTORE_STATE = 314,  
PROP_BACKUP_PREPARATION_TIME = 315,  
PROP_RESTORE_PREPARATION_TIME = 316,  
/* enumerations 317-323 are used in Addendum j to ANSI/ASHRAE  
135-2004 */  
PROP_USER_INFORMATION_REFERENCE = 317,  
PROP_USER_NAME = 318,  
PROP_USER_TYPE = 319,  
PROP_USES_REMAINING = 320,  
PROP_VENDOR_FORMAT_IDENTIFIER = 321,  
PROP_ZONE_FROM = 322,  
PROP_ZONE_TO = 323,  
/* enumerations 324-325 are used in Addendum i to ANSI/ASHRAE
```



```

135-2004 */
    PROP_BINARY_ACTIVE_VALUE = 324,
    PROP_BINARY_INACTIVE_VALUE = 325
    /* The special property identifiers all, optional, and required
*/
    /* are reserved for use in the ReadPropertyConditional and */
    /* ReadPropertyMultiple services or services not defined in this
standard. */
    /* Enumerated values 0-511 are reserved for definition by
ASHRAE. */
    /* Enumerated values 512-4194303 may be used by others subject to
the */
    /* procedures and constraints described in Clause 23. */
} BACNET_PROPERTY_ID;

```

E.2 BACnet Engineering Units Codes

BACnet engineering units codes may be found in your copy of the BACnet protocol specification, ANSI/ASHRAE Standard 135. That document is copyrighted, but the C enumeration shown below for reference is taken from open source code available under GPL at <http://sourceforge.net>, and provides essentially the same information (copyrighted by Steve Karg, licensed under GPL as noted at <http://sourceforge.net>).

```

typedef enum {
    /* Acceleration */
    UNITS_METERS_PER_SECOND_PER_SECOND = 166,
    /* Area */
    UNITS_SQUARE_METERS = 0,
    UNITS_SQUARE_CENTIMETERS = 116,
    UNITS_SQUARE_FEET = 1,
    UNITS_SQUARE_INCHES = 115,
    /* Currency */
    UNITS_CURRENCY1 = 105,
    UNITS_CURRENCY2 = 106,
    UNITS_CURRENCY3 = 107,
    UNITS_CURRENCY4 = 108,
    UNITS_CURRENCY5 = 109,
    UNITS_CURRENCY6 = 110,
    UNITS_CURRENCY7 = 111,
    UNITS_CURRENCY8 = 112,
    UNITS_CURRENCY9 = 113,
    UNITS_CURRENCY10 = 114,
    /* Electrical */
    UNITS_MILLIAMPERES = 2,
    UNITS_AMPERES = 3,
    UNITS_AMPERES_PER_METER = 167,
    UNITS_AMPERES_PER_SQUARE_METER = 168,
    UNITS_AMPERE_SQUARE_METERS = 169,
    UNITS_FARADS = 170,
    UNITS_HENRYS = 171,
    UNITS_OHMS = 4,
    UNITS_OHM_METERS = 172,

```

```
UNITS_MILLIOHMS = 145,  
UNITS_KILOHMS = 122,  
UNITS_MEGOHMS = 123,  
UNITS_SIEMENS = 173, /* 1 mho equals 1 siemens */  
UNITS_SIEMENS_PER_METER = 174,  
UNITS_TESLAS = 175,  
UNITS_VOLTS = 5,  
UNITS_MILLIVOLTS = 124,  
UNITS_KILOVOLTS = 6,  
UNITS_MEGAVOLTS = 7,  
UNITS_VOLT_AMPERES = 8,  
UNITS_KILOVOLT_AMPERES = 9,  
UNITS_MEGAVOLT_AMPERES = 10,  
UNITS_VOLT_AMPERES_REACTIVE = 11,  
UNITS_KILOVOLT_AMPERES_REACTIVE = 12,  
UNITS_MEGAVOLT_AMPERES_REACTIVE = 13,  
UNITS_VOLTS_PER_DEGREE_KELVIN = 176,  
UNITS_VOLTS_PER_METER = 177,  
UNITS_DEGREES_PHASE = 14,  
UNITS_POWER_FACTOR = 15,  
UNITS_WEBERS = 178,  
/* Energy */  
UNITS_JOULES = 16,  
UNITS_KILOJOULES = 17,  
UNITS_KILOJOULES_PER_KILOGRAM = 125,  
UNITS_MEGAJOULES = 126,  
UNITS_WATT_HOURS = 18,  
UNITS_KILOWATT_HOURS = 19,  
UNITS_MEGAWATT_HOURS = 146,  
UNITS_BTUS = 20,  
UNITS_KILO_BTUS = 147,  
UNITS_MEGA_BTUS = 148,  
UNITS_THERMS = 21,  
UNITS_TON_HOURS = 22,  
/* Enthalpy */  
UNITS_JOULES_PER_KILOGRAM_DRY_AIR = 23,  
UNITS_KILOJOULES_PER_KILOGRAM_DRY_AIR = 149,  
UNITS_MEGAJOULES_PER_KILOGRAM_DRY_AIR = 150,  
UNITS_BTUS_PER_POUND_DRY_AIR = 24,  
UNITS_BTUS_PER_POUND = 117,  
/* Entropy */  
UNITS_JOULES_PER_DEGREE_KELVIN = 127,  
UNITS_KILOJOULES_PER_DEGREE_KELVIN = 151,  
UNITS_MEGAJOULES_PER_DEGREE_KELVIN = 152,  
UNITS_JOULES_PER_KILOGRAM_DEGREE_KELVIN = 128,  
/* Force */  
UNITS_NEWTON = 153,  
/* Frequency */  
UNITS_CYCLES_PER_HOUR = 25,  
UNITS_CYCLES_PER_MINUTE = 26,  
UNITS_HERTZ = 27,  
UNITS_KILOHERTZ = 129,  
UNITS_MEGAHERTZ = 130,
```

```
UNITS_PER_HOUR = 131,  
/* Humidity */  
UNITS_GRAMS_OF_WATER_PER_KILOGRAM_DRY_AIR = 28,  
UNITS_PERCENT_RELATIVE_HUMIDITY = 29,  
/* Length */  
UNITS_MILLIMETERS = 30,  
UNITS_CENTIMETERS = 118,  
UNITS_METERS = 31,  
UNITS_INCHES = 32,  
UNITS_FEET = 33,  
/* Light */  
UNITS_CANDELAS = 179,  
UNITS_CANDELAS_PER_SQUARE_METER = 180,  
UNITS_WATTS_PER_SQUARE_FOOT = 34,  
UNITS_WATTS_PER_SQUARE_METER = 35,  
UNITS_LUMENS = 36,  
UNITS_LUXES = 37,  
UNITS_FOOT_CANDLES = 38,  
/* Mass */  
UNITS_KILOGRAMS = 39,  
UNITS_POUNDS_MASS = 40,  
UNITS_TONS = 41,  
/* Mass Flow */  
UNITS_GRAMS_PER_SECOND = 154,  
UNITS_GRAMS_PER_MINUTE = 155,  
UNITS_KILOGRAMS_PER_SECOND = 42,  
UNITS_KILOGRAMS_PER_MINUTE = 43,  
UNITS_KILOGRAMS_PER_HOUR = 44,  
UNITS_POUNDS_MASS_PER_SECOND = 119,  
UNITS_POUNDS_MASS_PER_MINUTE = 45,  
UNITS_POUNDS_MASS_PER_HOUR = 46,  
UNITS_TONS_PER_HOUR = 156,  
/* Power */  
UNITS_MILLIWATTS = 132,  
UNITS_WATTS = 47,  
UNITS_KILOWATTS = 48,  
UNITS_MEGAWATTS = 49,  
UNITS_BTUS_PER_HOUR = 50,  
UNITS_KILO_BTUS_PER_HOUR = 157,  
UNITS_HORSEPOWER = 51,  
UNITS_TONS_REFRIGERATION = 52,  
/* Pressure */  
UNITS_PASCALS = 53,  
UNITS_HECTOPASCALS = 133,  
UNITS_KILOPASCALS = 54,  
UNITS_MILLIBARS = 134,  
UNITS_BARS = 55,  
UNITS_POUNDS_FORCE_PER_SQUARE_INCH = 56,  
UNITS_CENTIMETERS_OF_WATER = 57,  
UNITS_INCHES_OF_WATER = 58,  
UNITS_MILLIMETERS_OF_MERCURY = 59,  
UNITS_CENTIMETERS_OF_MERCURY = 60,  
UNITS_INCHES_OF_MERCURY = 61,
```

```
/* Temperature */
UNITS_DEGREES_CELSIUS = 62,
UNITS_DEGREES_KELVIN = 63,
UNITS_DEGREES_KELVIN_PER_HOUR = 181,
UNITS_DEGREES_KELVIN_PER_MINUTE = 182,
UNITS_DEGREES_FAHRENHEIT = 64,
UNITS_DEGREE_DAYS_CELSIUS = 65,
UNITS_DEGREE_DAYS_FAHRENHEIT = 66,
UNITS_DELTA_DEGREES_FAHRENHEIT = 120,
UNITS_DELTA_DEGREES_KELVIN = 121,
/* Time */
UNITS_YEARS = 67,
UNITS_MONTHS = 68,
UNITS_WEEKS = 69,
UNITS_DAYS = 70,
UNITS_HOURS = 71,
UNITS_MINUTES = 72,
UNITS_SECONDS = 73,
UNITS_HUNDREDTHS_SECONDS = 158,
UNITS_MILLISECONDS = 159,
/* Torque */
UNITS_NEWTON_METERS = 160,
/* Velocity */
UNITS_MILLIMETERS_PER_SECOND = 161,
UNITS_MILLIMETERS_PER_MINUTE = 162,
UNITS_METERS_PER_SECOND = 74,
UNITS_METERS_PER_MINUTE = 163,
UNITS_METERS_PER_HOUR = 164,
UNITS_KILOMETERS_PER_HOUR = 75,
UNITS_FEET_PER_SECOND = 76,
UNITS_FEET_PER_MINUTE = 77,
UNITS_MILES_PER_HOUR = 78,
/* Volume */
UNITS_CUBIC_FEET = 79,
UNITS_CUBIC_METERS = 80,
UNITS_IMPERIAL_GALLONS = 81,
UNITS_LITERS = 82,
UNITS_US_GALLONS = 83,
/* Volumetric Flow */
UNITS_CUBIC_FEET_PER_SECOND = 142,
UNITS_CUBIC_FEET_PER_MINUTE = 84,
UNITS_CUBIC_METERS_PER_SECOND = 85,
UNITS_CUBIC_METERS_PER_MINUTE = 165,
UNITS_CUBIC_METERS_PER_HOUR = 135,
UNITS_IMPERIAL_GALLONS_PER_MINUTE = 86,
UNITS_LITERS_PER_SECOND = 87,
UNITS_LITERS_PER_MINUTE = 88,
UNITS_LITERS_PER_HOUR = 136,
UNITS_US_GALLONS_PER_MINUTE = 89,
/* Other */
UNITS_DEGREES_ANGULAR = 90,
UNITS_DEGREES_CELSIUS_PER_HOUR = 91,
UNITS_DEGREES_CELSIUS_PER_MINUTE = 92,
```

```
UNITS_DEGREES_FAHRENHEIT_PER_HOUR = 93,  
UNITS_DEGREES_FAHRENHEIT_PER_MINUTE = 94,  
UNITS_JOULE_SECONDS = 183,  
UNITS_KILOGRAMS_PER_CUBIC_METER = 186,  
UNITS_KW_HOURS_PER_SQUARE_METER = 137,  
UNITS_KW_HOURS_PER_SQUARE_FOOT = 138,  
UNITS_MEGAJOULES_PER_SQUARE_METER = 139,  
UNITS_MEGAJOULES_PER_SQUARE_FOOT = 140,  
UNITS_NO_UNITS = 95,  
UNITS_NEWTON_SECONDS = 187,  
UNITS_NEWTONS_PER_METER = 188,  
UNITS_PARTS_PER_MILLION = 96,  
UNITS_PARTS_PER_BILLION = 97,  
UNITS_PERCENT = 98,  
UNITS_PERCENT_OBSCURATION_PER_FOOT = 143,  
UNITS_PERCENT_OBSCURATION_PER_METER = 144,  
UNITS_PERCENT_PER_SECOND = 99,  
UNITS_PER_MINUTE = 100,  
UNITS_PER_SECOND = 101,  
UNITS_PSI_PER_DEGREE_FAHRENHEIT = 102,  
UNITS_RADIANS = 103,  
UNITS_RADIANS_PER_SECOND = 184,  
UNITS_REVOLUTIONS_PER_MINUTE = 104,  
UNITS_SQUARE_METERS_PER_NEWTON = 185,  
UNITS_WATTS_PER_METER_PER_DEGREE_KELVIN = 189,  
UNITS_WATTS_PER_SQUARE_METER_DEGREE_KELVIN = 141,  
    ; /* Enumerated values 0-255 are reserved for definition by  
ASHRAE. */  
    /* Enumerated values 256-65535 may be used by others subject to  
*/  
    /* the procedures and constraints described in Clause 23. */  
    /* The last enumeration used in this version is 189. */  
    MAX_UNITS = 190  
} BACNET_ENGINEERING_UNITS;
```




Appendix F Using Wireshark for Trouble Shooting

F.1 Hardware Requirements for Ethernet

There are no particular hardware requirements regarding the PC you run Wireshark on. Basically anything running any version of Windows can run Wireshark. There are also Linux and Mac versions.

The "hardware requirement" that is of most concern is the means of connecting to the network. We typically just connect everything Ethernet to a switch and don't worry about it. However, switches are really unmanaged routers, and they filter traffic. Therefore, your PC will not see traffic passing back and forth between two other devices that are not the PC. In order to see that network traffic using Wireshark, you need to come up with the right kind of network connection.

If your PC itself is one end of the network conversation you wish to capture, for example when running the Network Discovery Tool, then Wireshark will capture all network traffic to and from the PC however connected. It is when your PC wants to simply "eavesdrop" that you run into problems with the network switch.

A while back, 10BaseT hubs were common. A 10BaseT hub is not as smart as a switch and does not filter traffic. If you have an old 10BaseT hub collecting dust somewhere, you now have a new use for it. It will let Wireshark see all traffic from the PC that goes between any other devices connected to that 10BaseT hub. Beware of devices that call themselves "hubs" but support 100BaseT connections. These are switches.

Since manufacturers of hubs decided nobody should have a use for them anymore, they are generally out of production. Finding a 10BaseT hub for sale is not easy (try eBay). But there are other alternatives.

One means of monitoring network traffic is to get a managed switch that supports "port mirroring". One such device we have tested is the TP-LINK model TL-SG105E. Setting it up requires utility software (provided with the switch) and takes a little effort to get configured. But once configured, it works well without any further monkeying around. And it is inexpensive.

The other means of monitoring traffic is with the use of a device made specifically for use with Wireshark. The "SharkTap" provides two connections for the network pass-through, and a third "tap" connection where you connect your PC running Wireshark. There is no configuration required. It is the simplest way to monitor network traffic,

and it is a current production item available on Amazon (as of 2020).



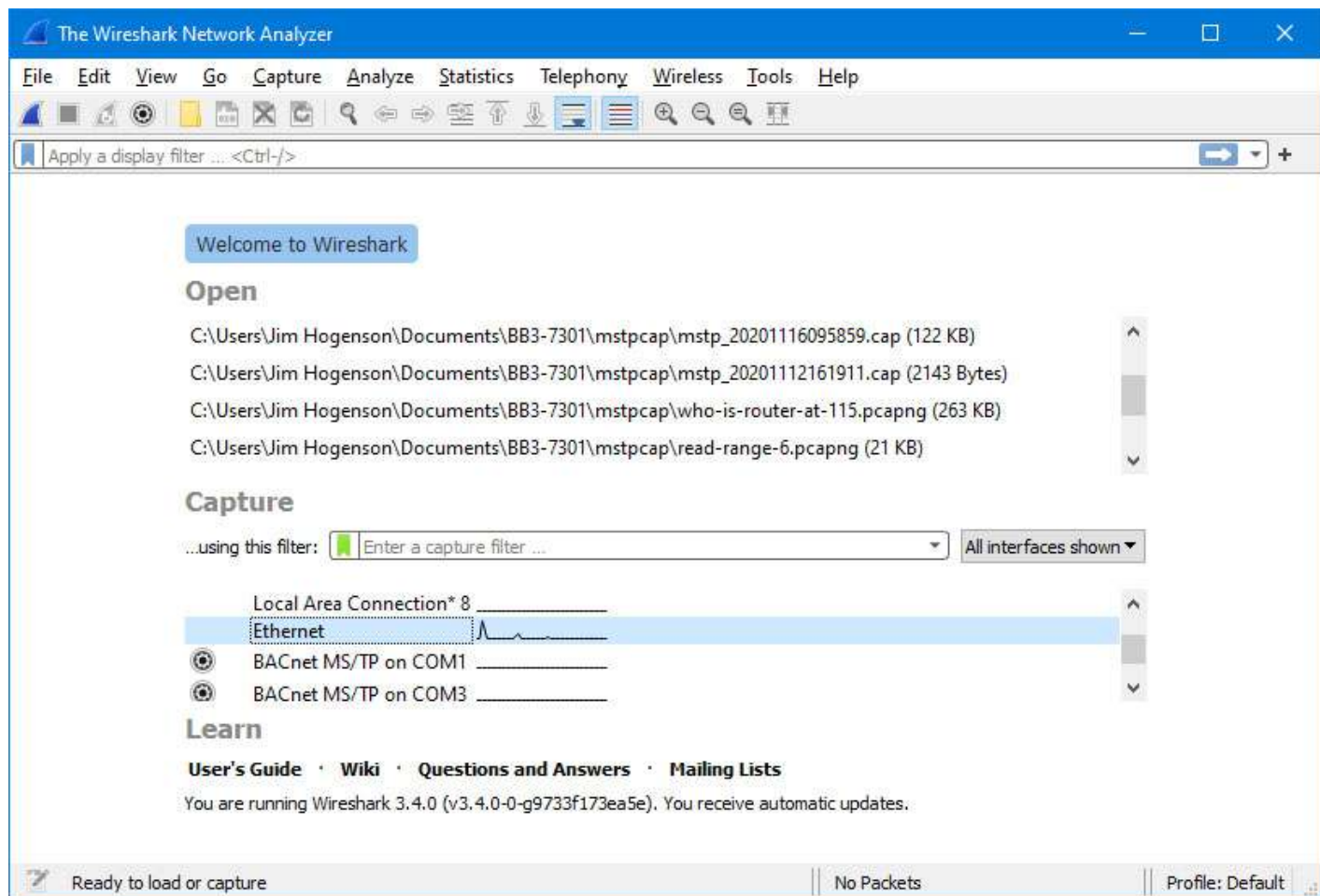
F.2 Hardware Requirements for MS/TP

Wireshark can be used in two ways with MS/TP. You can use the Control Solutions MTX002 to capture packets and then analyze them with Wireshark. This is talked about on the Control Solutions web site, on the MTX002 page. More recent versions of Wireshark now also support live capture of MS/TP using a standard RS-485 to RS-232 adapter with a standard serial COM port on a PC. A generic USB to RS-485 serial port adapter also works.

Live capture of MS/TP in Wireshark requires the support of an external capture application. You can find instructions and that application here: <https://steve.kargs.net/bacnet/bacnet-mstp-wireshark-live-capture/>. The live capture tool is provided on SourceForge.net, but Steve Karg's instructions are helpful.

F.3 Example of Using Wireshark

Using Wireshark is fairly easy. Get a copy at www.wireshark.org and install it. Once installed, running it is straight forward. As of version 3.4.0 of Wireshark, the startup screen looks like the following. Double click on Ethernet, or BACnet MS/TP on COMx, to start capturing network traffic. If you have multiple network connections, they will all be listed. Be sure to select the one that represents the correct connection.

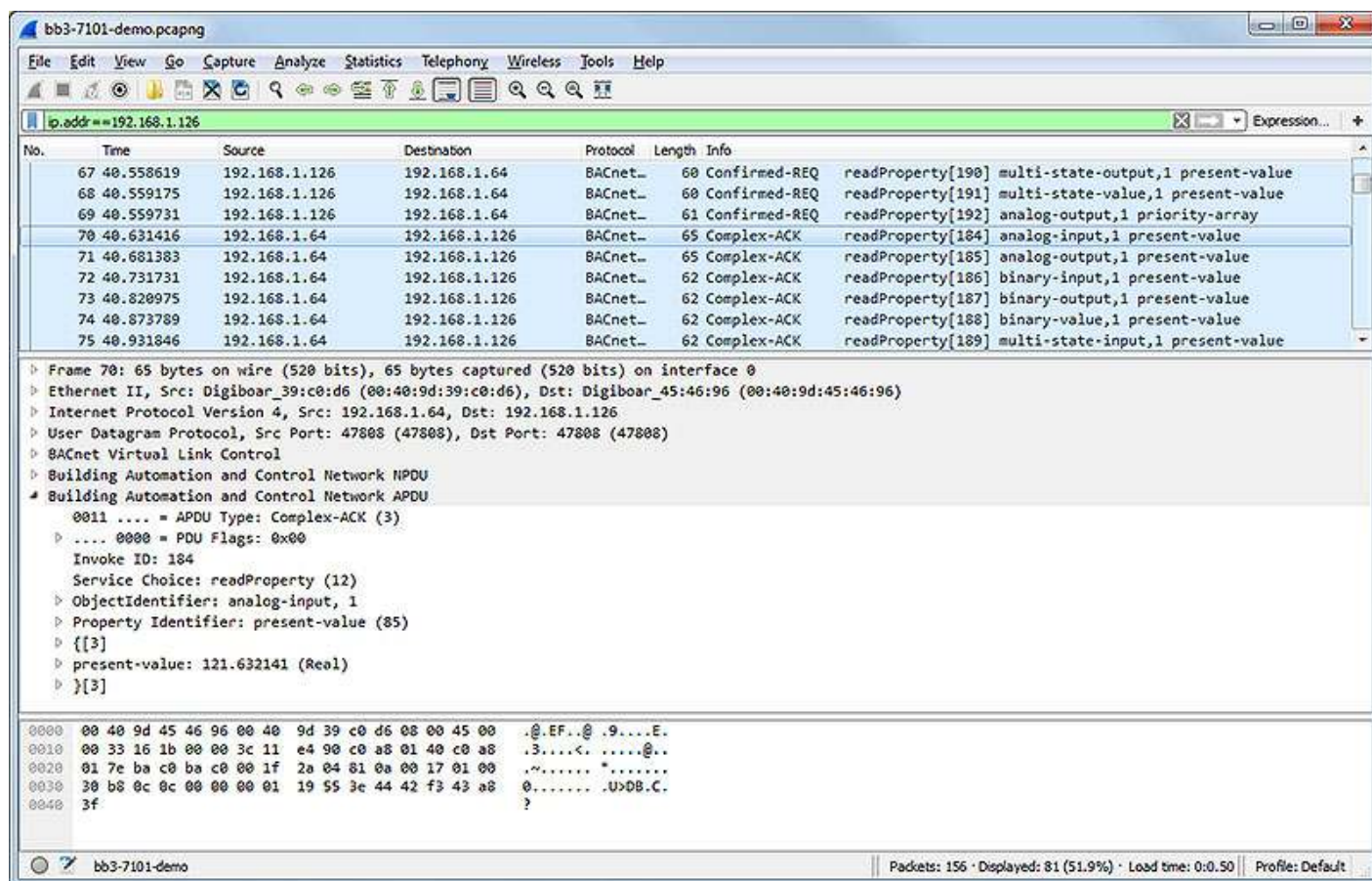


The screen will look something like the example below once Wireshark starts collecting data. Click the red icon in the toolbar to stop capturing traffic. Control Solutions technical support will often ask for a copy of the Wireshark data when a network issue seems evident. You can save a copy of all of the network traffic captured under the File menu, and you will generally save it to a .pcap or .pcapng file. A Wireshark log with .pcap extension can be posted directly as an attachment in support tickets while .pcapng needs to be zipped first.

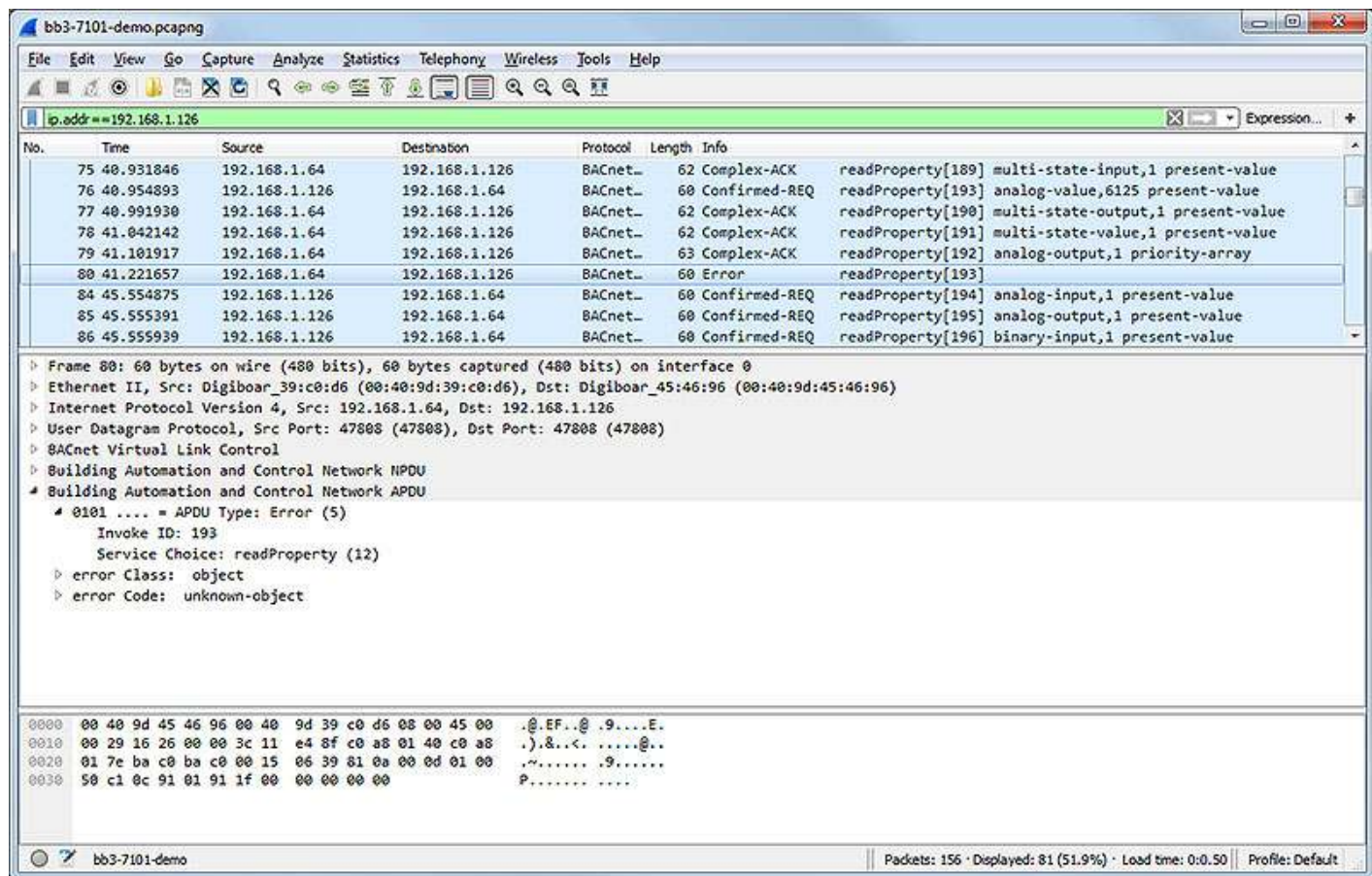
The screen shot below shows Wireshark capturing BACnet IP traffic between the local BACnet client and another BACnet device. If you click on a packet, the details of that packet will be displayed in the lower part of the screen. You can expand the tree view to see further detail.

A lot of times you will see a lot of network traffic that is not of interest to you. You can filter network traffic to only display traffic to/from the device you are interested in. Do this by entering "ip.addr==192.168.1.126" in the Filter window as illustrated below. (Substitute your own device's IP address.)

The example illustrated here is a Complex-ACK, or in other words, reply to a Read Property request.



This next example shows what an error reply will look like. This error resulted in the example screen shot in Section 5.4 of this User Guide.



The above examples showed filtering traffic by IP address. You can also filter simply by "bacnet" to see only BACnet traffic.

Wireshark capture of BACnet traffic on an Ethernet interface. The packet list shows a series of BACnet-APDU messages between 192.168.1.112 and 192.168.1.115. The selected packet (No. 36) is expanded to show the BACnet-APDU details, including PDU Type (Confirmed-REQ), PDU Flags, Max Response Segments, Size of Maximum ADPU, Invoke ID (115), Service Choice (readProperty), ObjectIdentifier (analog-output, 1), and Property Identifier (present-value). The packet bytes are shown in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
23	3.878362	192.168.1.112	192.168.1.115	BACnet-APDU	59	Confirmed-REQ readProperty[115] analog-output,1 present-value
24	3.878371	192.168.1.112	192.168.1.115	BACnet-APDU	59	Confirmed-REQ readProperty[115] analog-output,1 present-value
25	3.985160	192.168.1.115	192.168.1.112	BACnet-APDU	65	Complex-ACK readProperty[115] analog-output,1 present-value
27	5.070242	192.168.1.112	192.168.1.115	BACnet-APDU	59	Confirmed-REQ readProperty[116] analog-output,1 present-value
28	5.070250	192.168.1.112	192.168.1.115	BACnet-APDU	59	Confirmed-REQ readProperty[116] analog-output,1 present-value
29	5.082294	192.168.1.115	192.168.1.112	BACnet-APDU	65	Complex-ACK readProperty[116] analog-output,1 present-value
31	6.206372	192.168.1.112	192.168.1.115	BACnet-APDU	59	Confirmed-REQ readProperty[117] analog-output,1 present-value
32	6.206381	192.168.1.112	192.168.1.115	BACnet-APDU	59	Confirmed-REQ readProperty[117] analog-output,1 present-value
33	6.303937	192.168.1.115	192.168.1.112	BACnet-APDU	65	Complex-ACK readProperty[117] analog-output,1 present-value
34	7.318252	192.168.1.112	192.168.1.115	BACnet-APDU	59	Confirmed-REQ readProperty[118] analog-output,1 present-value
35	7.318261	192.168.1.112	192.168.1.115	BACnet-APDU	59	Confirmed-REQ readProperty[118] analog-output,1 present-value
36	7.328670	192.168.1.115	192.168.1.112	BACnet-APDU	65	Complex-ACK readProperty[118] analog-output,1 present-value

Internet Protocol Version 4, Src: 192.168.1.112, Dst: 192.168.1.115
 User Datagram Protocol, Src Port: 47808, Dst Port: 47808
 BACnet Virtual Link Control
 Building Automation and Control Network NPDU
 Building Automation and Control Network APDU
 0000 = APDU Type: Confirmed-REQ (0)
 0000 = PDU Flags: 0x0
 .000 = Max Response Segments accepted: Unspecified (0)
 0011 = Size of Maximum ADPU accepted: Up to 480 octets (fits in an ARCNET frame) (3)
 Invoke ID: 115
 Service Choice: readProperty (12)
 ObjectIdentifier: analog-output, 1
 Property Identifier: present-value (85)

0000 00 40 9d 43 e8 3c a4 bb 6d a2 b9 f6 08 00 45 00 .@C<- m---E
 0010 00 2d d8 ee 00 00 80 11 00 00 c0 a8 01 70 c0 a8 --...p
 0020 01 73 ba c0 ba c0 00 19 84 5e 81 0a 00 11 01 04 .s.....^
 0030 00 03 73 0c 0c 00 40 00 01 19 55 ..s...@..L

Building Automation and Control Network APDU (bacapp), 11 bytes

Packets: 53 · Displayed: 12 (22.6%) · Dropped: 0 (0.0%) Profile: Default

The above examples show BACnet IP traffic on the Ethernet port. The following is an example of Wireshark capturing MS/TP traffic.

The image shows a Wireshark packet capture window titled "net-discover-1.pcapng". The main pane displays a list of captured packets. Packet 616 is selected, and its details pane shows the structure of a BACnet-APDU. The packet list shows the following data:

No.	Time	Source	Destination	Protocol	Length	Info
609	7.866680	0x2a	0x50	BACnet	8	BACnet MS/TP Poll For Master
610	7.896599	0x2a	0x00	BACnet	8	BACnet MS/TP Token
611	7.905575	0x00	0x13	BACnet	8	BACnet MS/TP Poll For Master
612	7.935495	0x00	0x15	BACnet	8	BACnet MS/TP Token
613	7.942477	0x15	0x16	BACnet	8	BACnet MS/TP Token
614	7.952453	0x16	0x1f	BACnet	8	BACnet MS/TP Token
615	7.972398	0x1f	0xff	BACnet-APDU	34	Unconfirmed-REQ i-Am device,545
616	7.984365	0x1f	0x15	BACnet-APDU	23	Confirmed-REQ readProperty[210] analog-value,1 p
617	7.999325	0x15	0x1f	BACnet-APDU	29	Complex-ACK readProperty[210] analog-value,1 p
618	8.017278	0x1f	0xff	BACnet-APDU	34	Unconfirmed-REQ i-Am device,176
619	8.032238	0x1f	0xff	BACnet-APDU	34	Unconfirmed-REQ i-Am device,16010
620	8.047199	0x1f	0xff	BACnet-APDU	34	Unconfirmed-REQ i-Am device,16006

The details pane for packet 616 shows the following structure:

- > Frame 616: 23 bytes on wire (184 bits), 23 bytes captured (184 bits) on interface wireshark_extcap2100, id 0
- > BACnet MS/TP, Src (31), Dst (21), BACnet Data Expecting Reply
- > Building Automation and Control Network NPDU
- ✓ Building Automation and Control Network APDU
 - 0000 = APDU Type: Confirmed-REQ (0)
 - > 0000 = PDU Flags: 0x0
 - .000 = Max Response Segments accepted: Unspecified (0)
 - 0011 = Size of Maximum ADPU accepted: Up to 480 octets (fits in an ARCNET frame) (3)
 - Invoke ID: 210
 - Service Choice: readProperty (12)
 - > ObjectIdentifier: analog-value, 1
 - > Property Identifier: present-value (85)

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0000 55 ff 05 15 1f 00 0d 21 01 04 00 03 d2 0c 0c 00 U.....! .....
0010 80 00 01 19 55 53 e0 ....LS
```

The status bar at the bottom indicates: Building Automation and Control Network APDU (bacapp), 11 bytes | Packets: 2916 · Displayed: 2916 (100.0%) | Profile: Default



Appendix G SSL Certificates for Secure Web (HTTPS)

The secure web server (HTTPS) requires SSL certificates in order to establish secure connections. The HTTPS certificates are only required if HTTPS is enabled on the Network configuration page in the Babel Buster BB3-7302/MX-73.

G.1 X.509 Auto-Certificate Generation

The Babel Buster BB3-7302/MX-73 Gateway will automatically generate X.509 certificates if no external certificates are found or could not be loaded correctly. These will be generated one time and saved in the Flash file system for subsequent reuse. When the self-generated X.509 certificates are in use, this will be indicated at the bottom of the Network configuration page.

A screenshot of the Network configuration page. It has a dark teal background with white text. At the top, 'Web Server' is followed by two checked checkboxes: 'HTTPS Enabled (on 443)' and 'HTTP Enabled'. Below this, 'HTTP Port' is set to '80' in a text box, with '(default 80)' to its right. A 'Set Ports' button is to the right of the text box. Below that, 'MIB Offset' is set to '0' in a text box. Further down, 'FTP Server' is followed by a checked checkbox and the word 'Enabled'. At the bottom left, 'MAC Address: 00:40:9D:45:47:10' is displayed. At the bottom right, 'System Uptime: 0,00:01:39' is displayed. At the very bottom, it says 'HTTPS certificate status: Using self-generated X.509'.

If there is a need to delete the self-generated certificates, you can do so by logging in via FTP. Change directory to /FLASH0, then to .cfg. The two certificate files that were self-generated are ssl.cert and ssl.key.

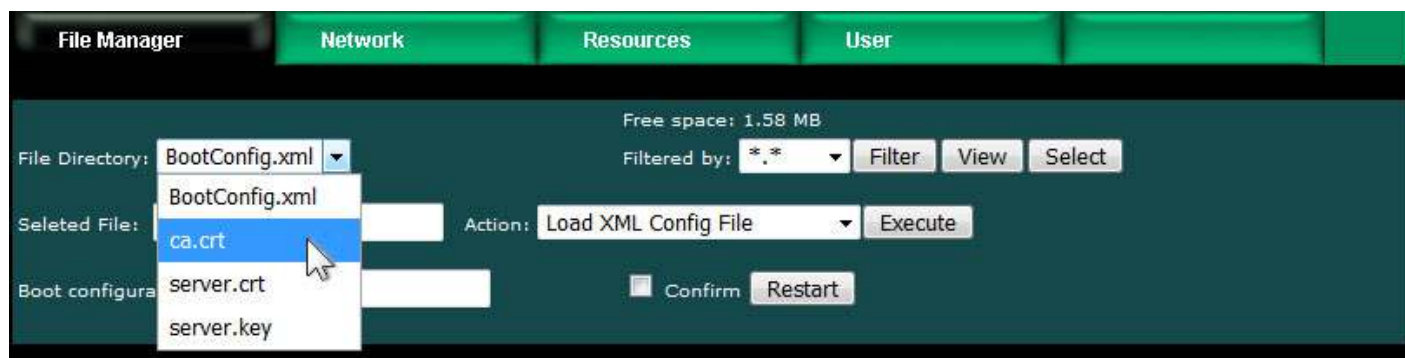
```

C:\Users\Jim Hogenson\My Documents\config files>ftp
ftp> open 192.168.1.120
Connected to 192.168.1.120.
220 NET+OS 7.5.2.2 FTP server ready.
User (192.168.1.120:(none)): root
331 User root OK, send password.
Password:
230 Password OK.
ftp> cd /FLASH0
250 Directory is changed
ftp> dir .cfg
200 PORT command Ok.
150 File Listing Follows in ASCII mode
-rwlrwl--- 1 noone      group2 447      Dec 31 1969 ssl.cert
-rwlrwl--- 1 noone      group2 465      Dec 31 1969 ssl.key
226 Transfer complete.
ftp: 119 bytes received in 0.11Seconds 1.09Kbytes/sec.
ftp>

```

G.2 External Certificates

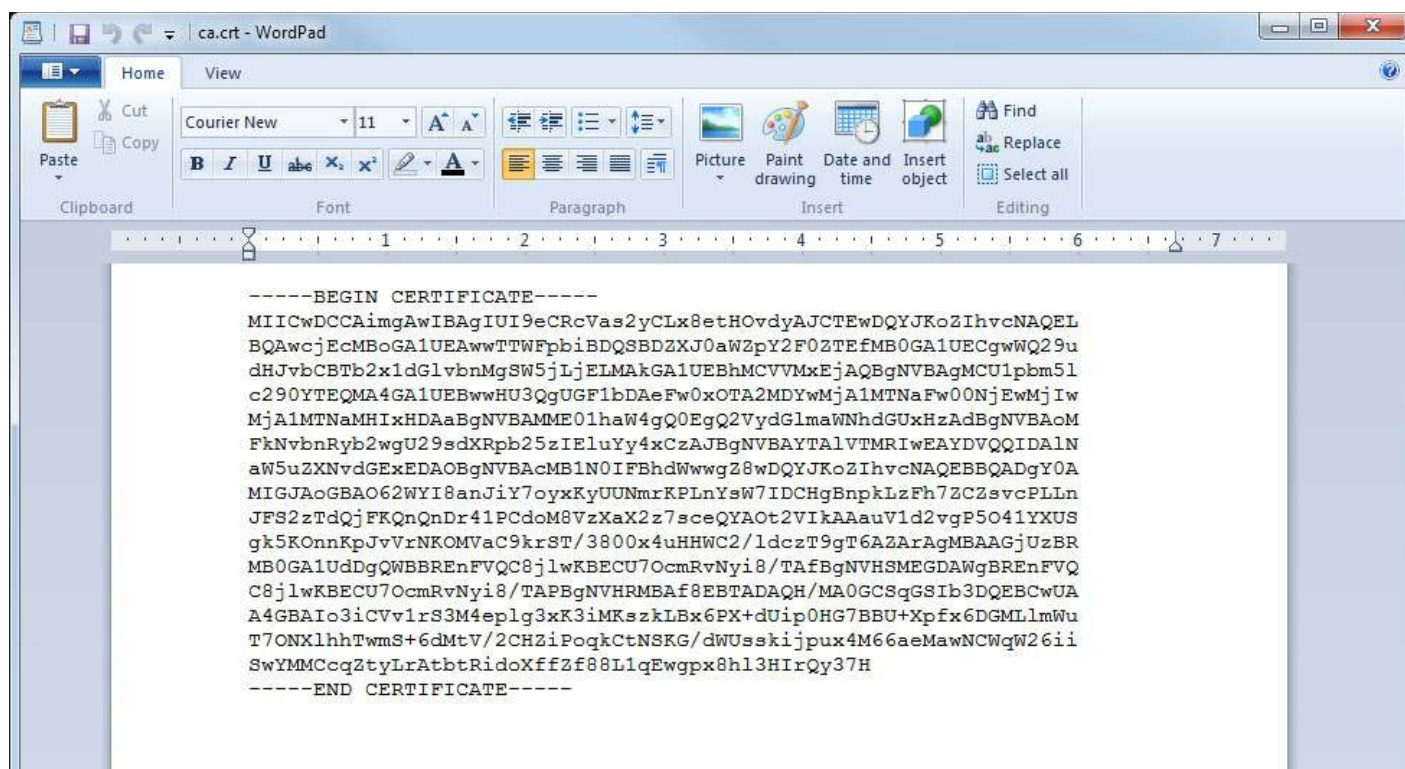
There are three certificates that you must generate and upload to use SSL certificates other than the self-generated X.509 certificates.



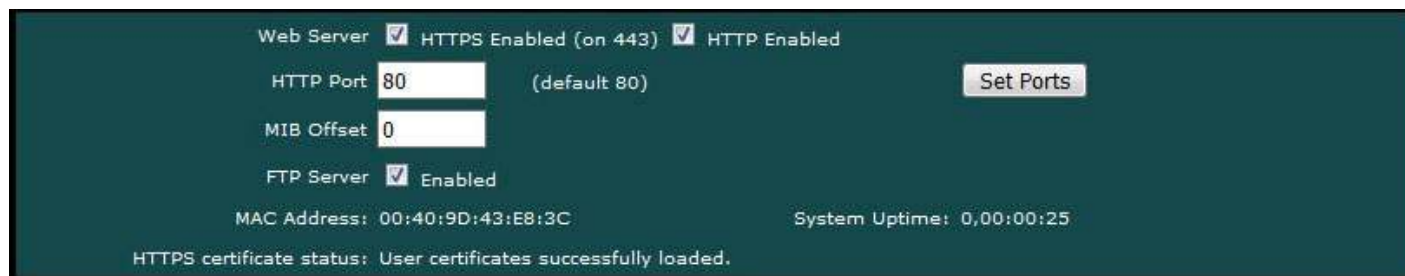
The required certificates are as follows, and must use exactly these names.

ca.crt	CA Root certificate in PEM format
server.crt	Server certificate in PEM format
server.key	Server private key in PEM format

The content of each certificate file will look something like the screen shot below. If you require external certificates for your secure web server, the requirement was likely imposed by your IT department. They should be able to provide the necessary certificates for you. For globally accessed use, the Root CA would come from somebody like GoDaddy or DigiCert (formerly Symantec).



If external certificates were loaded successfully, that will be indicated at the bottom of the Network configuration page.



G.3 Certificate Generation Script (Linux)

The art and science of generating SSL certificates is beyond the scope of this document. An example SSL certificate generation script is provided here as a reference.

The following script, run on a Linux system with OpenSSL installed, will generate the three required SSL certificate files. It will generate a number of intermediate files as well - you don't need to upload them. Replace references to Control Solutions in this script with your own company name.

```
#!/bin/bash
echo hello
# This will create some self signed certs, using one master CA.
#
# these can be the webserver DNS name, or an IP address, however you
access
# the resource, this needs to match.
```



```
if [ -z "$1" ] || [ -z "$2" ]; then
echo 'Usage: gen.sh <server-name> <client-name>'
echo ' <server-name> and <client-name> can be IP addresses'
echo ' or DNS names.'
exit 1
fi
SNAME=$1
CNAME=$2
#
# Bits for strength, 1024, 2048, 4096, etc.. (suggest 2k or 4k for web
servers)
BITS=1024
#
# HASH - Options are sha256, sha512, sha1, md5
HASH="sha256"
SN=`date +%Y%m%d%H%M%S`
#####
# below is the entry for the CRL
# Do not use http://www.csimn.com/crl.pem for production keys and
certificates
# cat <<EOF >> extensions.cnf
# [ extensions_section ]
# crlDistributionPoints = URI:http://www.csimn.com/crl.pem
#
# basicConstraints = CA:FALSE
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment
# subjectAltName = DNS:${SNAME},IP:${SNAME}
# EOF
#####
#####
# first, lets generate some private keys...
openssl genrsa -out server.key ${BITS}
openssl genrsa -out client.key ${BITS}
# ok, and now the MAIN CA
openssl req -x509 -${HASH} -nodes -days 10000 -newkey rsa:${BITS} -keyout
ca.key -out ca.crt -subj "/CN=Main CA Certificate/O=Control Solutions
Inc./C=US/ST=Minnesota/L=St Paul"
#####
#
# Create a CSR for both server and client
# Replace these values with one appropriate for your organization
openssl req -out server.csr -key server.key -new -subj "/CN=${SNAME}
/O=Control Solutions Inc./C=US/ST=Minnesota/L=St Paul"
openssl req -out client.csr -key client.key -new -subj "/CN=${CNAME}
/O=Control Solutions Inc./C=US/ST=Minnesota/L=St Paul"
#
#
#####
# Sign the keys with the CA
openssl x509 -req -days 3650 -in server.csr -CA ca.crt -CAkey ca.key
-set_serial ${SN}01 -out server.crt -${HASH}
openssl x509 -req -days 3650 -in client.csr -CA ca.crt -CAkey ca.key
-set_serial ${SN}02 -out client.crt -${HASH}
```

```
# Create a windows file to import the client keys if needed in this
format
openssl pkcs12 -export -clcerts -in client.crt -inkey client.key -out
client.p12
# Create the client keys as a complete pem file if needed in this format
openssl pkcs12 -in client.p12 -out client-full.pem -clcerts
# mv -f server.key svrkey.pem
# mv -f server.crt svrcert.pem
# mv -f client.key clntkey.pem
# mv -f client.crt clntcert.pem
# cp -f ca.crt cacert.pem
###
# cleanup
# rm -f client.csr server.csr
#DLS 20160420
echo '*****'
echo '* WARNING: Do not use this script to generate production *'
echo '* keys and certificates. This script is for *'
echo '* demonstration purposes only. *'
echo '*****'
```

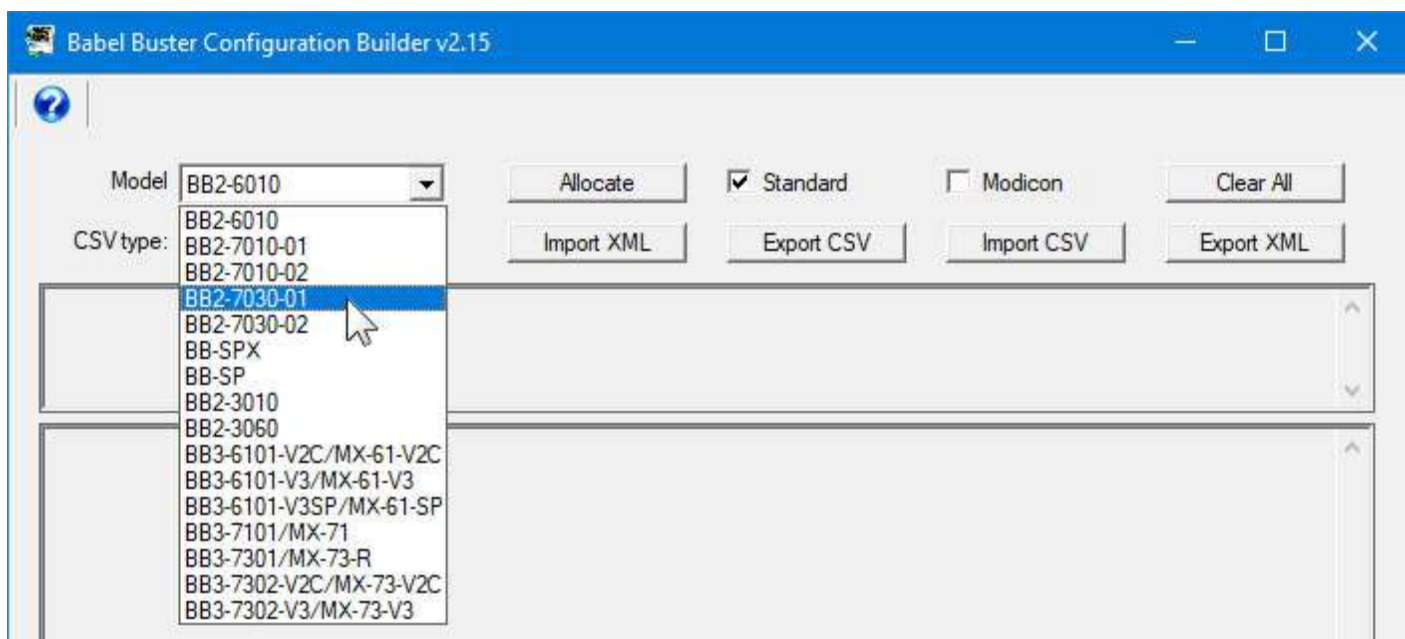


Appendix H Converting Older XML Files

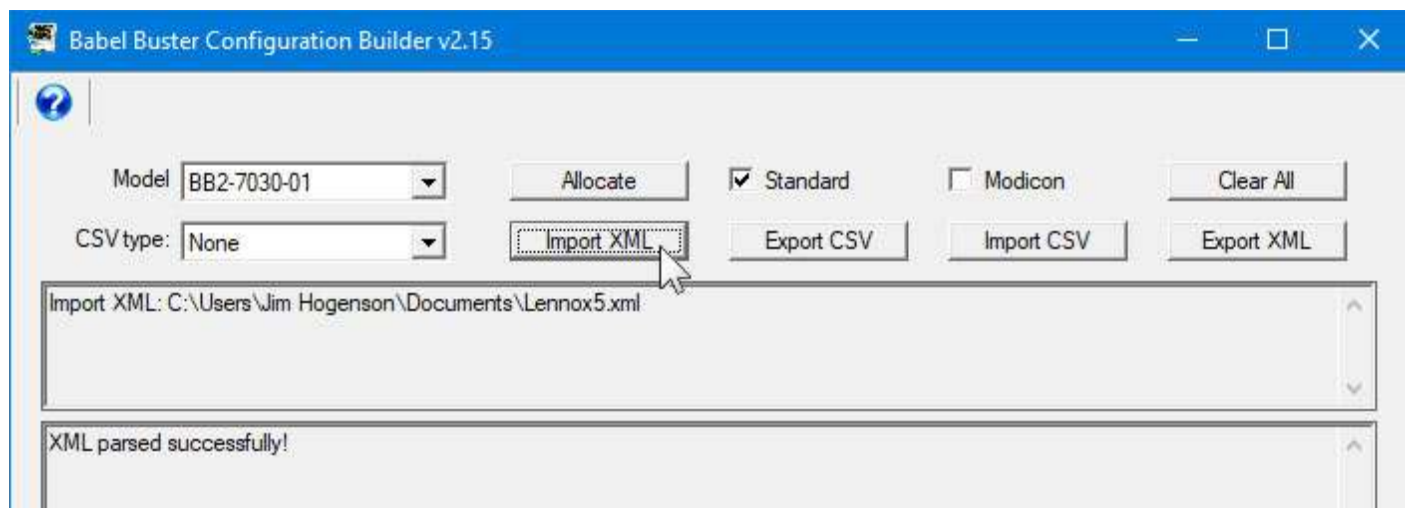
H.1 BB2-7030 to BB3-7302/MX-73 Conversion

You cannot load a BB2-7030 configuration XML file into the BB3-7302/MX-73, but you can easily convert it into a BB3-7302/MX-73 XML file using the Babel Buster Configuration Builder tool available under the Support/Tools link at csimn.com.

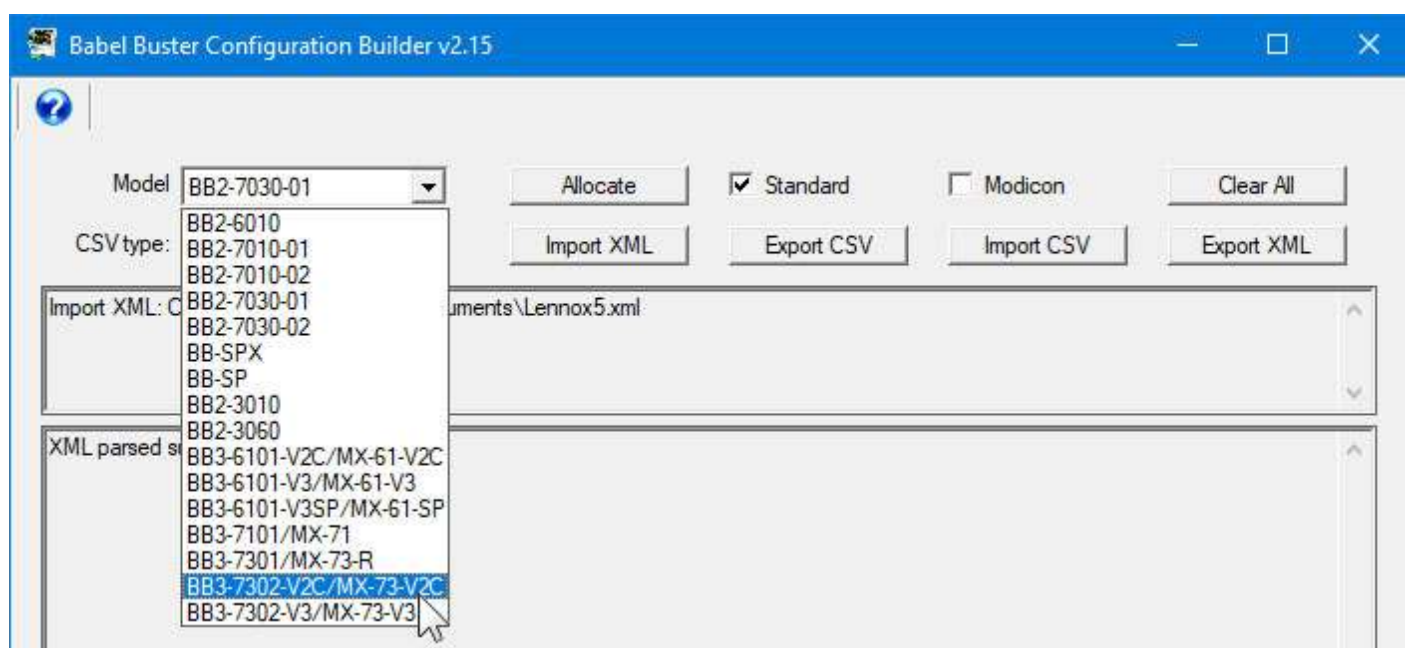
Converting the file is simple. Start by selecting the BB2-7030 model.



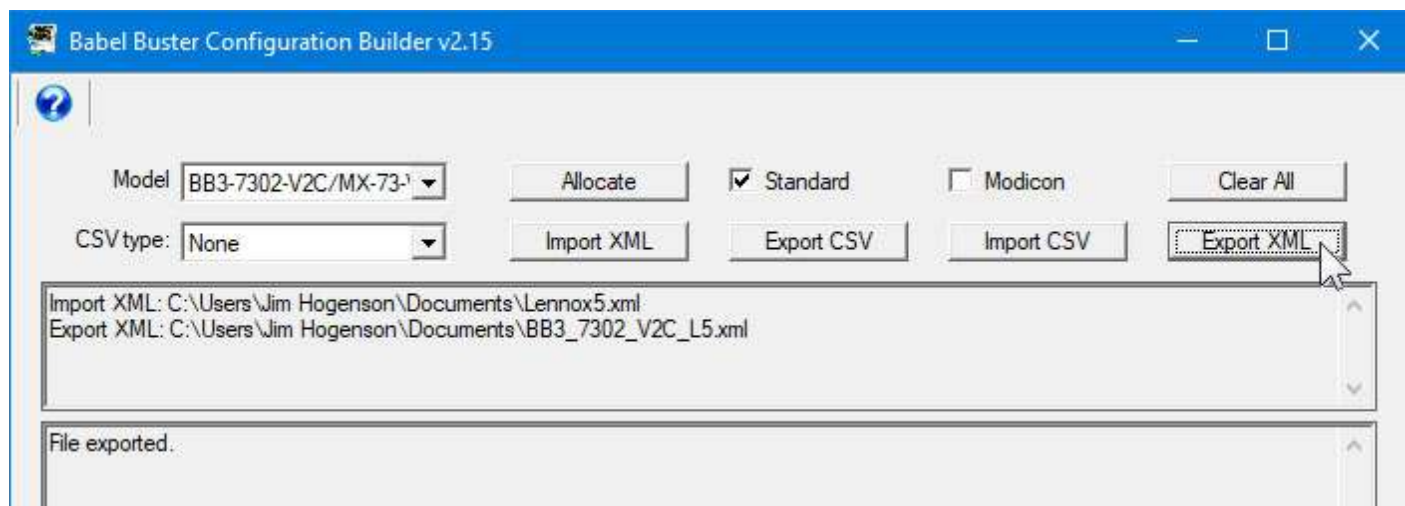
Click the Import XML button. The Windows file dialog window will open. Select your XML file from wherever you saved it on your PC. Upon opening the XML file, it will read the file into the configuration builder tool.



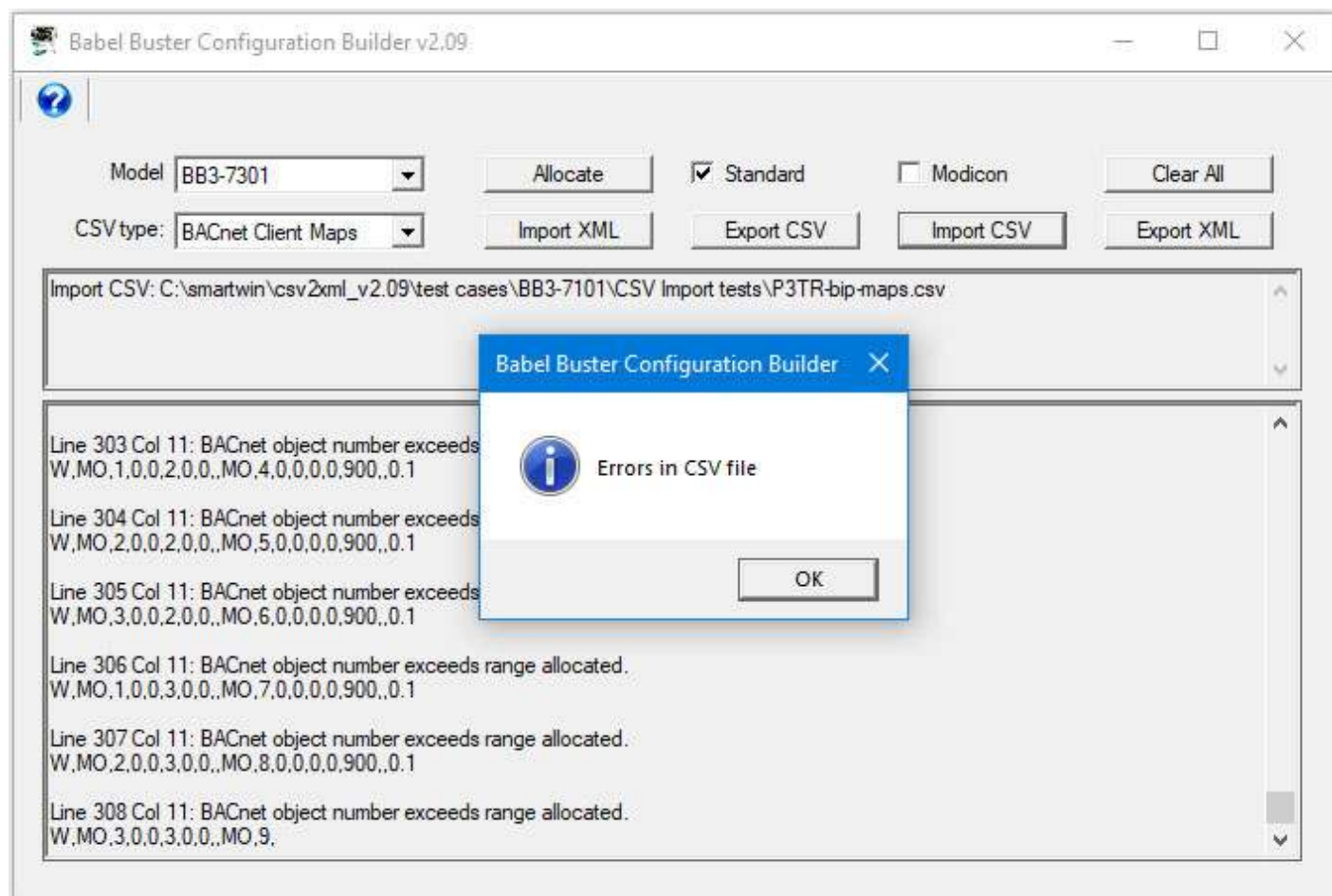
Now go back to the model list and switch to BB3-7302/MX-73.



Click the Export XML button. The Windows file dialog window will open again, this time wanting you to provide a name for the new XML file you are about to create. Choose a different name than the one you imported so that you don't lose your original file. Once the export is complete, you may now upload this new file to your BB3-7302/MX-73 gateway.



If things go terribly wrong, your screen may look something like the one below instead of the good examples above. If you are importing a BACnet related XML file and see "Invalid object...", the first thing to check is to click the Allocate button and be sure you do have objects allocated that are sufficient for the file being imported.



H.2 CSV File Export

The BB3-7302/MX-73 will not export CSV files directly. To create CSV files from an existing BB3-7302/MX-73 configuration, download a copy of the file to your PC, and then use the Babel Buster Configuration Builder to export CSV files as needed. To save

a copy of the XML configuration file to your PC, select the file and click the View button on the File Manager page in the BB3-7302/MX-73. Your browser will now display the XML file. **DO NOT** do a text copy/paste to try to create an XML file - doing so will result in an invalid file format that cannot be loaded again. You must use the browser's "save as" or "save page" function. The browser should default to wanting to save a file with a .xml suffix. If correctly saved on your PC, you should be able to double click on the saved file and it will result in opening the file automatically in your browser. It was saved correctly if the browser does not give any error messages when displaying the XML (which should now look exactly as it did when you first clicked the View button).

The Babel Buster Configuration Builder will export more types of CSV files than the BB3-7302/MX-73 itself will import. The most frequently used CSV files - namely read/write maps for the BACnet client - can be imported directly into the BB3-7302/MX-73. You do not generally have to import any Objects CSV file to start using objects - the objects are created automatically when the Resources page is updated. The only time you might be concerned about importing objects is if you have a long list of unique object names. In this case, import the CSV file into the configuration builder tool, and export as XML and then upload the XML file instead.

Importing device lists directly into the BB3-7302/MX-73 is also not supported. You generally only need to enter things like IP address once, and any one IP address can only be used one time. The effort to enter the IP addresses into the web UI of the BB3-7302/MX-73 is no greater than the effort to enter them into a spread sheet and then save and import. In fact, when it comes to setting up device lists, the CSV import would usually end up being more work. If you do have a device list in the form of a CSV file, import that into the configuration builder tool, export as XML (along with any other CSV files you import as part of the configuration), and then upload the resulting XML file to the BB3-7302/MX-73.