



## User Guide

Model VP4-0630, VP4-2330  
**ValuPoint 4**  
**Programmable I/O**  
**for BACnet MS/TP**

Rev. 1.2 – Oct. 2016

### ValuPoint User Guide Contents

#### [1 Overview](#)

- 1.1 How to Use This Guide
- 1.2 Important Safety Notice
- 1.3 Overview of the ValuPoint Devices
- 1.4 Warranty
- 1.5 Required License Information

#### [2 Installation and Connections](#)

- 2.1 Installing the configuration software
- 2.2 Installing the USB to MS/TP adapter
- 2.3 Connecting the ValuPoint
- 2.4 Connecting the USB Adapter
- 2.5 Indicators on the VP4-0630
- 2.6 Indicators on the VP4-2330
- 2.7 Indicators on the MTX002 USB to MS/TP Adapter

#### [3 Connect](#)

- 3.0 Do This for First Time Startup
- 3.1 MS/TP
- 3.2 IP
- 3.3 Work Offline
- 3.4 Auto Connect

#### [4 Who-Is](#)

- 4.0 Do This for First Time Startup
- 4.1 Send Who-Is
- 4.2 Refresh
- 4.3 Select Target
- 4.4 Get Device Info
- 4.5 Get Object List
- 4.6 Clear Who-Is Cache
- 4.7 Read All Config
- 4.8 Who-Is via IP Router

#### [5 Read/Write](#)

- 5.0 Skip This for First Time Startup
- 5.1 Read Property
- 5.2 Write Property

#### [6 Device Page](#)

- 6.0 Do This for First Time Startup
- 6.1 Get Info
- 6.2 Change Info
- 6.3 Reassign Device Instance
- 6.4 Get Counts
- 6.5 Recalculate and Reconfigure Counts
- 6.6 Reinit Device
- 6.7 File Read/Write (Device XML)

#### [7 I/O Point Assignments](#)

- 7.0 Skip This for First Time Startup
- 7.1 Assigning Physical I/O Points to BACnet Objects

## [8 Object Map Page](#)

- 8.0 Skip This for First Time Startup
- 8.1 Common Parameters
- 8.2 Physical Hardware I/O Parameters
- 8.3 MS/TP BACnet Client Parameters
- 8.4 Global Object
- 8.5 Insert, Add, Delete
- 8.6 Read/Write Device
- 8.7 File Read/Write (Object Map XML)

## [9 Object Map List](#)

- 9.0 Do This for First Time Startup
- 9.1 Read All, Write All
- 9.2 Select for Edit
- 9.3 File Read/Write (Object Map CSV)

## [10 PL/i Programming for Control](#)

- 10.0 Skip This for First Time Startup
- 10.1 Program Loading and Execution
- 10.2 Program Editing and Debugging
- 10.3 Program Capacity
- 10.4 Program States and Error Codes

## [11 View Data](#)

- 11.0 Skip This for First Time Startup
- 11.1 Read All
- 11.2 Data Definitions

## [12 Object Properties](#)

- 12.1 Data Object Properties (Analog, Binary, Multi-state)
- 12.2 Device Object Properties

## [13 Physical I/O Connections](#)

- 13.1 Connection of Inputs
- 13.2 Connection of Outputs
- 13.3 Connection of Power and Communications, VP4-0630
- 13.4 Connection of Power and Communications, VP4-2330

## [14 Trouble Shooting](#)

- 14.1 Error Codes, Reliability Codes
- 14.2 Auto-Reset Errors
- 14.3 MS/TP Trouble Shooting - Connecting the USB Adapter
- 14.4 MS/TP Trouble Shooting - Device Behavior
- 14.5 System Fault Indications

## [15 Editing a CSV Object Map File](#)

### [Appendix A – USB Driver Installation](#)

### [Appendix B – BACnet Property Type Codes](#)



# User Guide

## ValuPoint® 4

Model VP4-0630, VP4-2330  
Programmable I/O for MS/TP Networks  
Rev. 1.2 – Oct. 2016

© 2016 Control Solutions, Inc.

## 1 Overview

### 1.1 How to Use This Guide

Section 1 gives an overview of the ValuPoint programmable I/O device. Section 2 talks about installing the configuration software and connecting the ValuPoint. Sections 3 through 11 are guides for each of the tabs found on the screen of the configuration software. Sections 12 through 14, and Appendix A through C, are reference material.

The first subsection in each of sections 3 through 11 is titled "Do This" or "Skip This". The "Do This" sections are the minimum required actions necessary for connecting and syncing the device with the tool for the first time. You can then go back and study additional sections that follow "Do This" or go back and do the sections that initially said "Skip This".

### 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 Overview of the ValuPoint Devices

Control Solutions' ValuPoint® controllers are designed to be building blocks in a distributed control system where controllers communicate peer to peer via BACnet MS/TP. Each ValuPoint controller can access BACnet objects in other BACnet devices, both ValuPoint controllers and other manufacturer's devices. Remote objects are mapped to local objects that mirror the value of the remote object. The ValuPoint® Programmable I/O for BACnet MS/TP provides the standard profile BACnet Application Specific Controller (B-ASC).

All of the ValuPoint features available to local objects are also available to the remote mapped objects. This means the remote objects can be incorporated into the local i.CanDrawIt® program.

The user has the ability to assign object allocation as part of setup of the controller. Resources are allocated from a common pool, and the user has the ability to decrease the number of one type of object in order to increase the number of another type. Total object count will range from approximately 100 commandable objects (with full priority array) to 200 read/write objects. Object types include analog, binary, and multi-state variations of input, commandable output, and writable value.

#### Model VP4-0630

ValuPoint VP4-0630 is a scaled down, low cost version of the VP4-2330 featuring 2 universal inputs that may be configured as analog or discrete, 2 dedicated discrete inputs, and 2 Form A relay outputs.

#### Hardware Features of Model VP4-0630

- 2 Analog/universal inputs, software selectable types
  - 0-10VDC, discrete, pulse count
  - 10-bit A/D resolution
  - Non-volatile totalizing count inputs
- 2 Discrete Inputs
  - Dry contact closure to ground
  - Pulse count (10Hz) capability
  - Non-volatile totalizing count inputs
- 2 Discrete outputs
  - Form A relay
  - 2A @ 120VAC
  - 2A @ 30VDC
- BACnet MS/TP at 9600 to 76800 baud
- Hardened RS-485 port
- 128KB non-volatile EEPROM configuration file capacity
- 64K Flash for User Program
- ARM 32-bit processor, 512K Flash, 56K RAM
- Powered by 18-30VDC or 24VAC 50/60 Hz

- 0.2A @ 24VDC
- DIN rail mounting, 100mm H x 70mm W x 60mm D
- -40C to +70C, 5%-95% RH non-condensing
- Certifications: FCC, CE, UL 916

## Model VP4-2330

ValuPoint VP4-2330 features 14 universal context sensitive inputs. When configured as an analog input, the continuously self-calibrating sigma-delta converter produces 15-bit resolution with high noise immunity. When configured as a discrete input, the converter switches to 10-bit resolution at 1,000 samples per second for fast response. ValuPoint 4-2330 also includes 3 analog outputs, 6 Form A relay outputs, and pulse counting capability on all inputs.

## Hardware Features of Model VP4-2330

- I/O points and undedicated BACnet objects
  - Input objects: AI, BI, MSI
  - Commandable output objects: AO, BO, MSO
  - Writeable value objects: AV, BV, MSV
- Configurable object allocation for I/O and program use
- 14 Analog/universal inputs, software selectable types
  - 0-10VDC, thermistor, discrete, dry contact, pulse
  - 0.1% reference, up to 16-bit resolution
  - Continuously self-calibrating sigma-delta converter
  - Non-volatile totalizing count inputs  
(to 20Hz on all channels, to 5kHz on 2 channels)
- 3 Analog outputs
  - 4-20mA (0-20mA), 8-bit resolution, 500 ohm load max.
- 6 Discrete outputs
  - Form A relay, 5A @ 24VAC (120VAC industrial)
- Battery backed real time clock/calendar
- BACnet MS/TP at 9600 to 76800 baud
- Hardened RS-485 port
- 128KB non-volatile EEPROM configuration file capacity
- ARM 32-bit processor, 512K Flash, 96K RAM
- Powered by 18-30VDC or 24VAC 50/60 Hz
  - 0.3A @ 24VDC
- DIN rail mounting, 100mm H x 105mm W x 60mm D
- -40C to +70C, 5%-95% RH non-condensing
- Certifications: FCC, CE, UL 916

## Configuring the ValuPoint Programmable I/O Device

The ValuPoint is configured by writing various BACnet properties to the respective objects being configured. Each BACnet object, such as Analog Input, Analog Output, Binary Input, etc., includes properties defining which physical I/O point or remote

BACnet object should be mapped to this BACnet object. The usual mandatory object properties are included in each BACnet object found in the ValuPoint. There are additional proprietary object properties defining the physical I/O or BACnet client map parameters.

You can configure the ValuPoint using any BACnet client capable of reading and writing object properties. This process is streamlined in the ValuPoint configuration software tool available for free download at [www.csimn.com](http://www.csimn.com). The configuration tool provides easy-to-use templates you fill in, and then click a 'Write' button to send multiple properties to the ValuPoint at one time. Once configured, you can save your entire configuration to an XML format file on your PC for later re-use. To replicate the same configuration, simply load the XML file back into the configuration tool and click the 'Write All' button.

The ValuPoint configuration tool uses Control Solutions' MTX002 USB to MS/TP port adapter. The MTX002 allows connecting to MS/TP devices via the USB port on your PC. The MTX002 is intended primarily as an aid in configuring Control Solutions devices, but the tools do include general purpose BACnet object browsing capability usable with any MS/TP device.

The MTX002 operates as an MS/TP client device on the network, reading and writing properties in the device being configured as directed by the configuration tool applicable to the product being configured. As a client, the MTX002 can also read/write any standard BACnet properties in any MS/TP device using the BACnet diagnostic page of the software tool available for free download on the support pages.



MTX002 USB to MS/TP Adapter

## 1.4 Warranty

**This software and 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 or in the product(s) and/or the program(s) described in this documentation at any time. This product could include

software bugs, technical inaccuracies, typographical errors, and the like. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the software.

## 1.5 Required License Information

No licence is required for use of the ValuPoint configuration tool or programming software. The only licence requirements are that the following notices are provided.

The ValuPoint configuration and line programming tools include the SmartWin library (<http://smartwinlib.org>) under the following terms:

License agreement for SmartWin++ (BSD license)

Copyright (c) 2005, Thomas Hansen All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of the SmartWin++ nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

i.CanDrawIt includes, licensed under LGPL, TinyCAD, Copyright 1994-2009 Matt Pyne. Source code is available at <http://tinycad.sourceforge.net>. Open source products included under either GPL or LGPL include TinyCAD v2.70; Unicode/Font Conversions: iconv.dll version 1.9.0.0; PNG Image Support: libpng13.dll version 1.2.8.0; Image compression support: zlib1.dll version 1.2.1.0.



## 2 Installation and Connections

### 2.1 Installing the configuration software

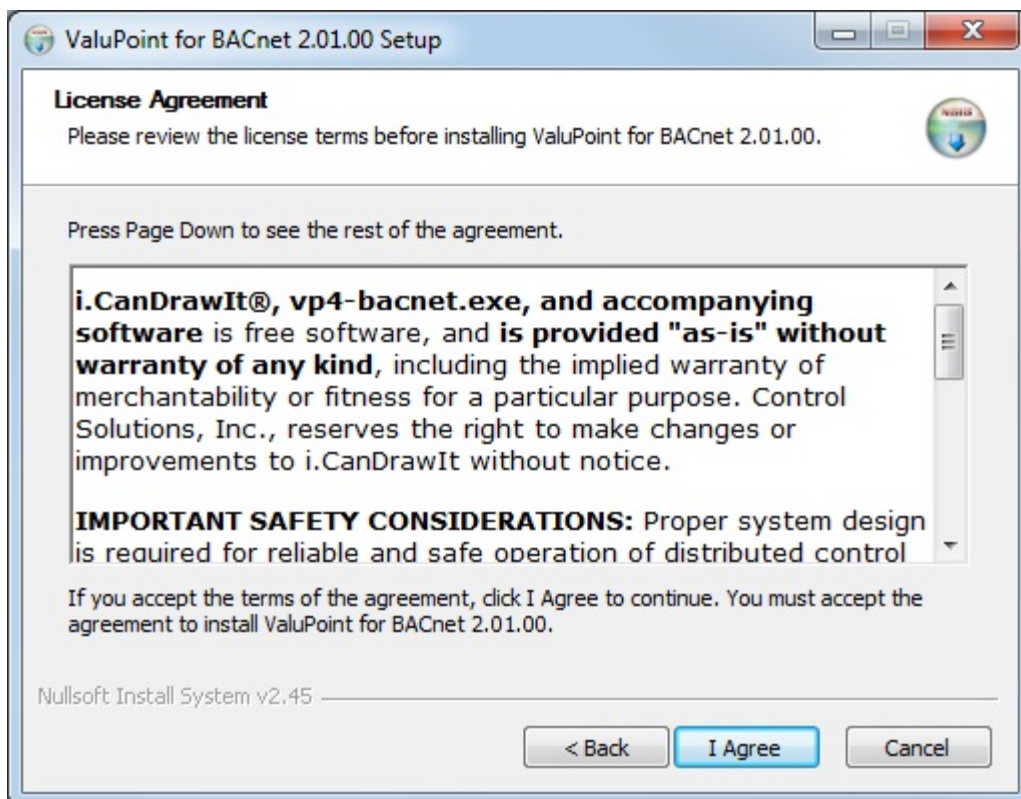
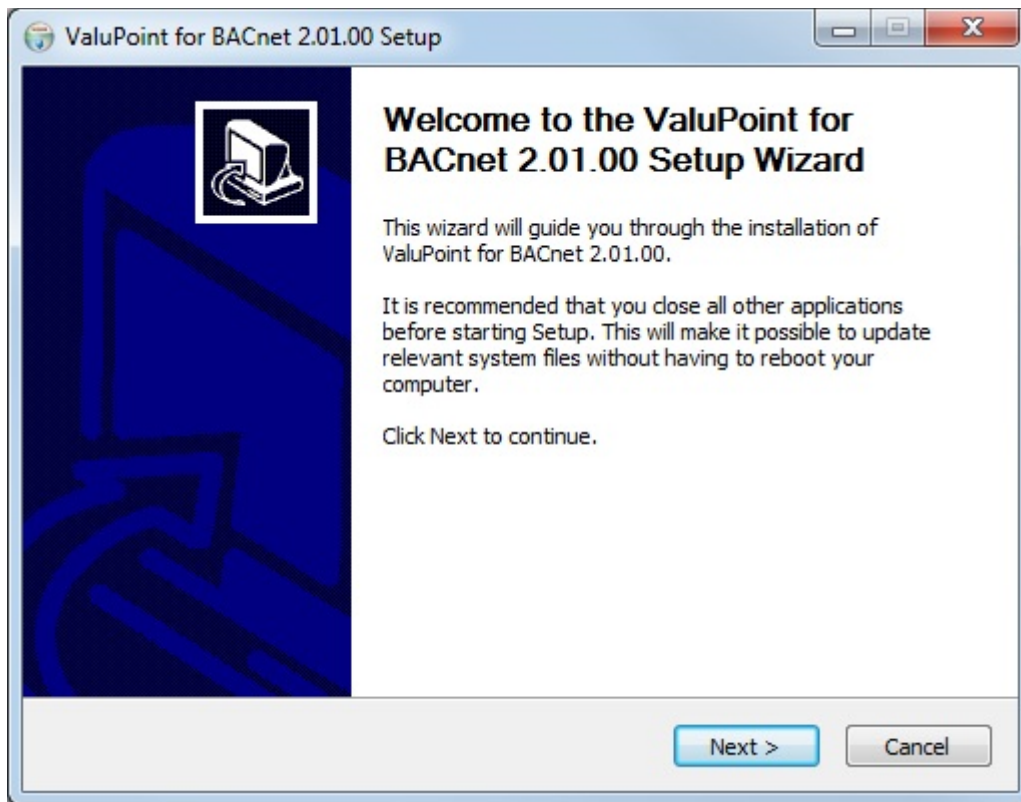
Look for the installer icons in the directory where you unzipped the download that got you to this document. The installer icons look like this:

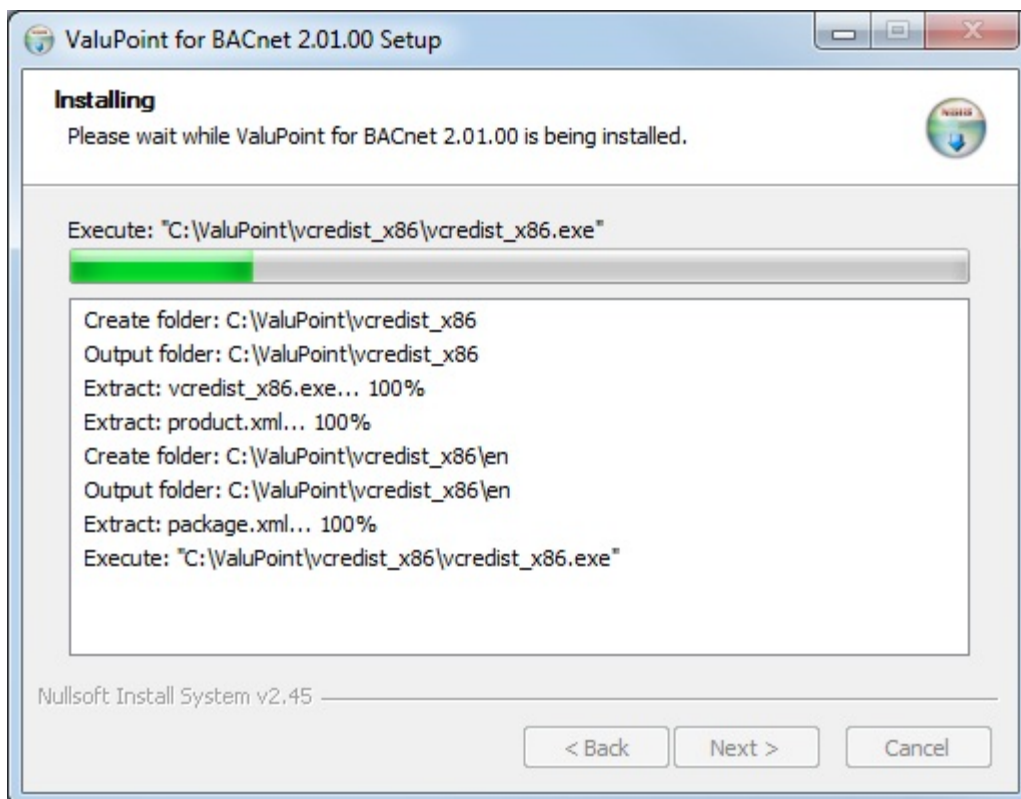
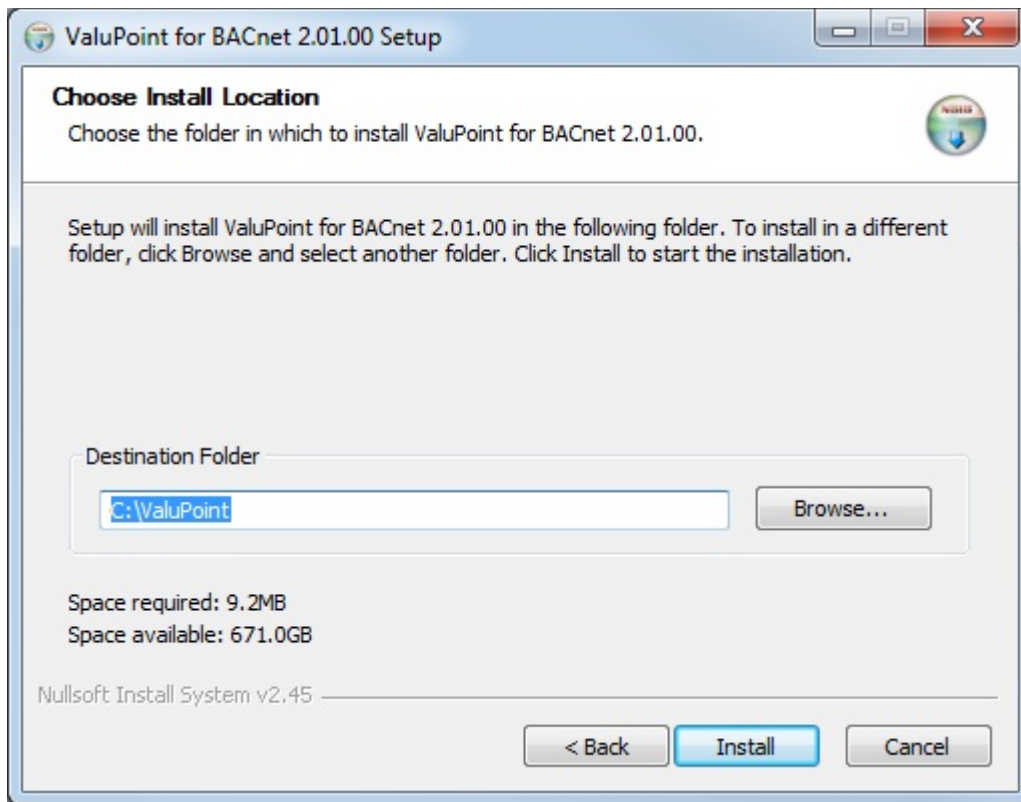


The installation is a 2-step installation. Install VP4-BACnet first. Then install i.CanDrawIt second.

Double click the icon to run the setup.exe. You will be questioned about whether to continue because Windows cannot verify the publisher of the software. Permit installation to continue. The sequence of installer screens include the following on Windows 7:

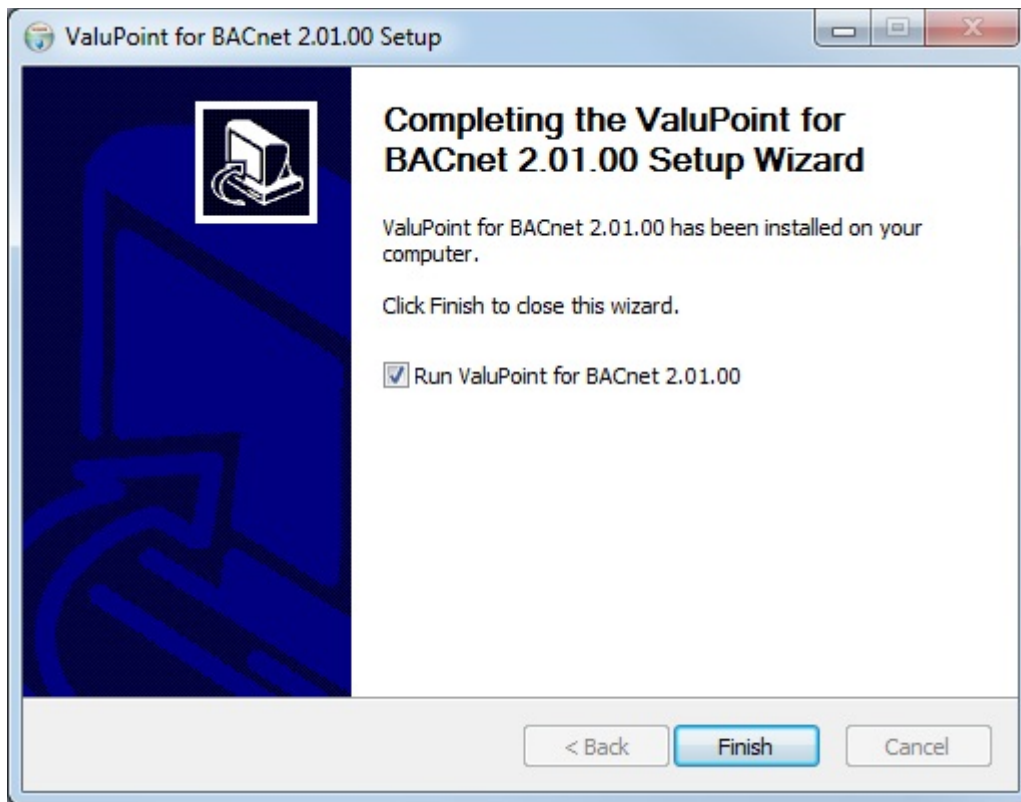




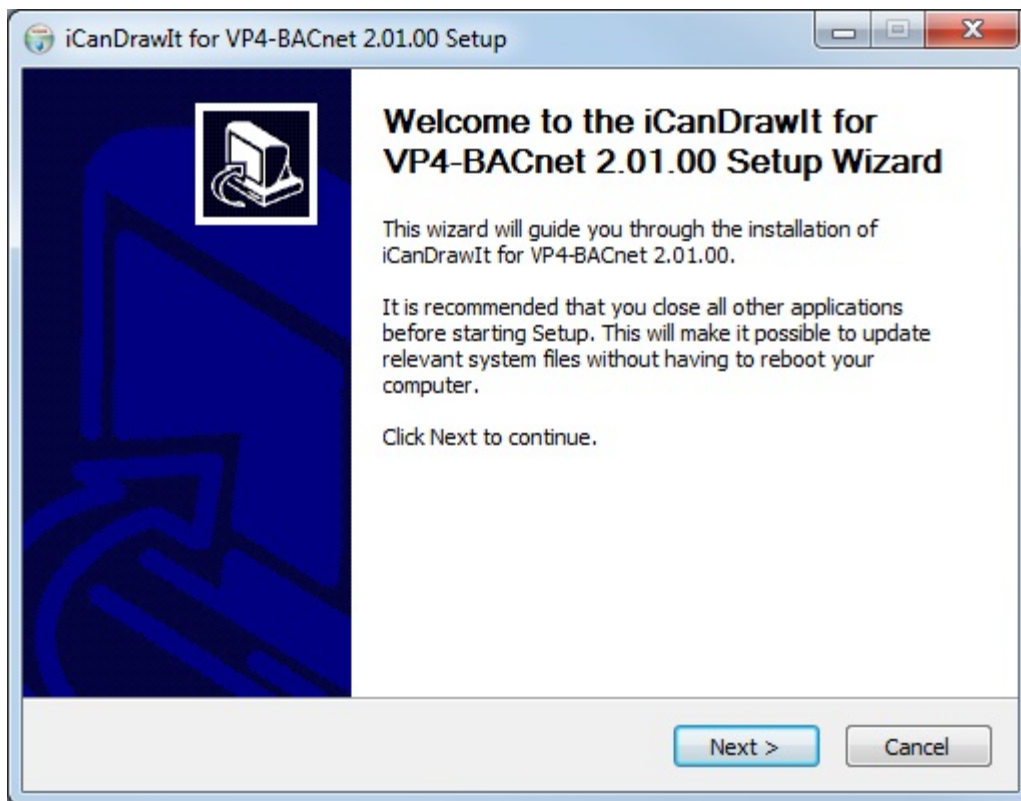


The installer will check to see whether Visual C++ support is already installed on your system, and install it if not. This is standard software provided by Microsoft.

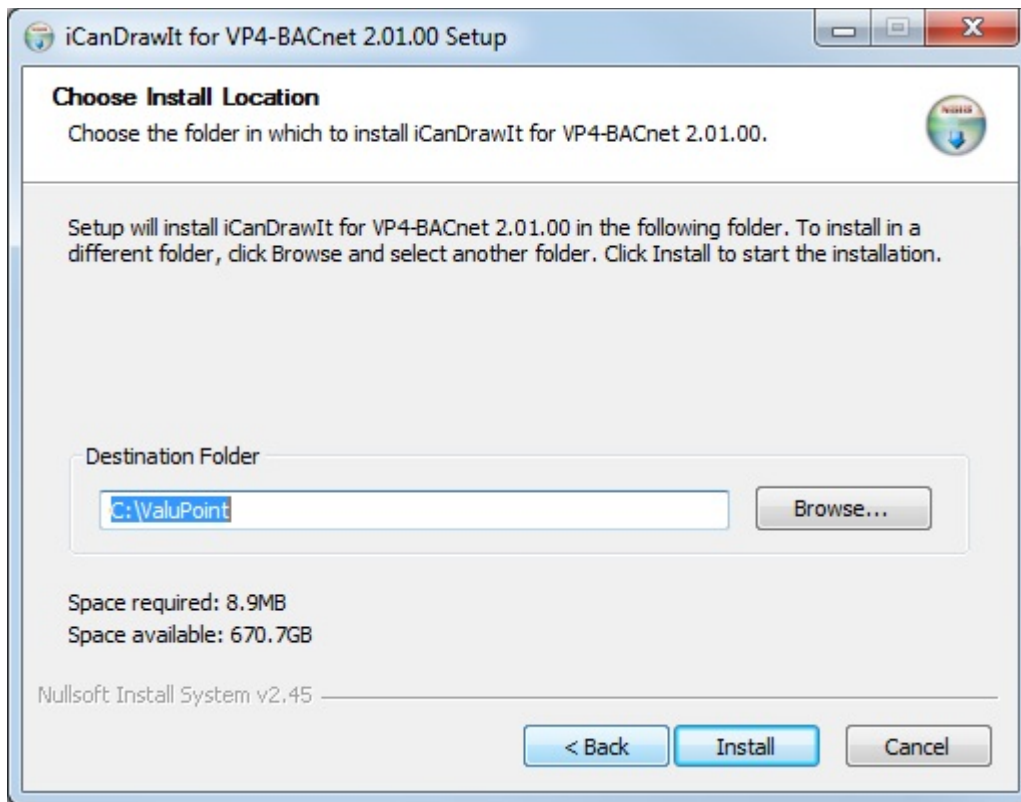
When you get to the "Finish" screen, you are ready to go.



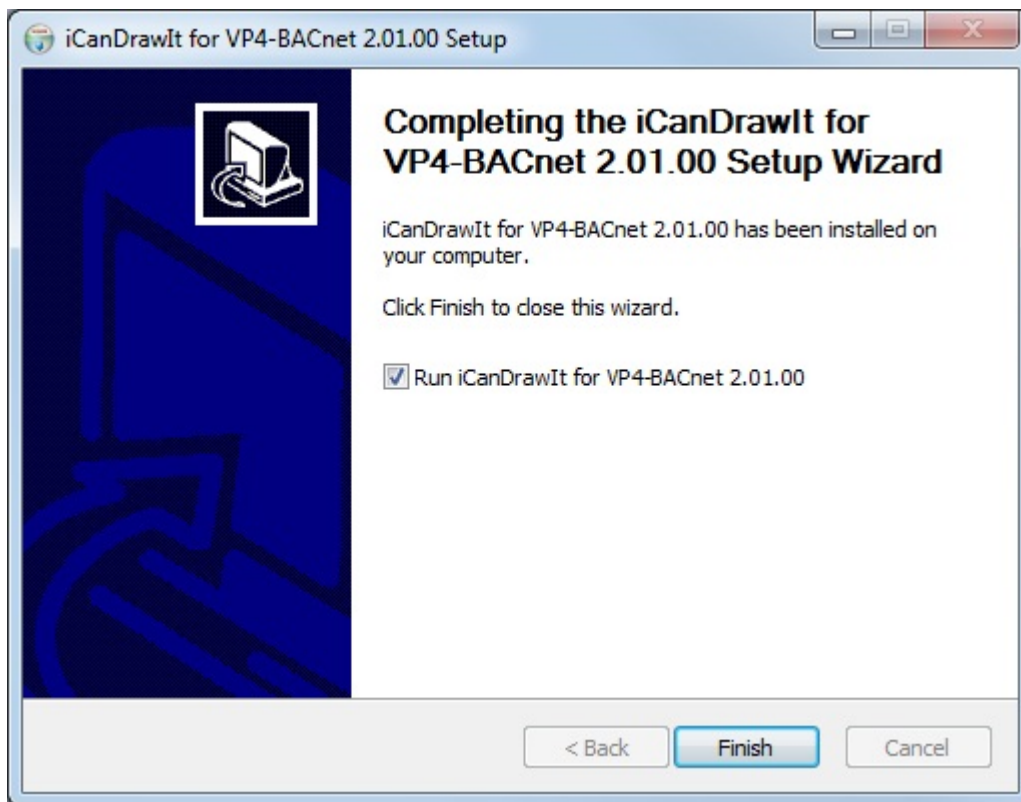
Next, proceed to install i.CanDrawIt. This part is optional. If you will not be using the ValuPoint as a programmable controller, you can skip this step. The first installer screen looks like this:



The installation directory should be the same directory that VP4-BACnet was installed into.



After a few more screens, you will get the familiar 'done' screen.



## 2.2 Installing the USB to MS/TP adapter

Install the USB adapter by plugging it into your PC. However, prior to doing so, you

should install the USB driver provided by Control Solutions. Complete instructions including step by step screen shots are provided in Appendix A.

The necessary driver was already included in Windows XP, but anything newer requires an explicitly installed driver. A properly signed and verified USB driver is provided for the MTX002.

### 2.3 Connecting the ValuPoint

Connecting Power: Connect +24VDC or 24VAC to the terminal marked POWER. Connect common to the terminal marked GND.

Connecting BACnet MS/TP: Connect RS-485 (EIA-485) data lines to NET+ and NET- next to POWER. Observe polarity of the data lines. Reverse polarity will be tolerated (i.e., not damage the unit) but will not function either. Connect RS-485 common and/or cable shield to one of the terminals marked GND as applicable.

Connecting I/O: Refer to section 13 for physical I/O connections.

### 2.4 Connecting the USB Adapter

Connect the NET+ and NET- terminals to the corresponding MS/TP terminals on the gateway (or to +/- on the MS/TP network). Connect GND to network common.



DO NOT connect power to the terminal block on the adapter. The adapter gets its power from the USB port of your PC. The adapter only draws a very small amount of power, well below the limit provided by standard USB ports.

### 2.5 Indicators on the VP4-0630 Programmable I/O



The LED indicators for the VP4-0630 are mounted on the circuit board inside the device, and are viewable through the vent slots in the case. The two blue LEDs are power indicators. There are two power supplies in the VP4-0630, and both are necessary for proper operation.

The four LEDs toward the bottom are status and communications - primarily used for communications. The green LED to the left indicates token passing. This LED flashes each time the token is passed. The red LED is the error indicator which flashes red when there is an error, such as request for an object that does not exist in the device. The yellow indicator flashes each time the VP4-0630 polls for master. The green LED to the right flashes each time a data request is replied to.

The yellow poll-for-master LED may flash quite frequently, or nearly never. How often the device polls for master depends on whether there is another master at some higher MAC address after the VP4-0630. If the next consecutive MAC address is in use by another device on the link, the poll for master will rarely flash. If the VP4-0630 has a low MAC address and there are few or no other devices with higher MAC addresses, the poll for master LED will be quite busy.

The token pass led (green LED on left) should always be flashing. If there is a small number of devices on the network, it will flash rapidly. The more devices there are on the network, the more slowly it will flash because there are more devices passing the token around. If it never flashes, there is a communication problem, such as baud rate mismatch.

The red LED is a communication error indicator most of the time. However, during restart it will be on solid. Immediately following startup, it may flash a fault code if a serious error has occurred.

## 2.6 Indicators on the VP4-2330 Programmable I/O



The LED indicators for the VP4-2330 are mounted on the circuit board inside the device, and are viewable through the vent slots in the case. The two blue LEDs are power indicators. There are two power supplies in the VP4-2330, and both are necessary for proper operation.

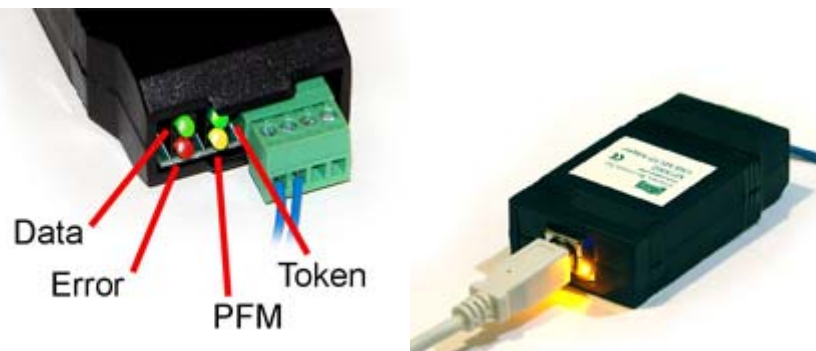
The four LEDs toward the bottom are status and communications - primarily used for communications. The green LED to the left indicates token passing. This LED flashes each time the token is passed. The red LED is the error indicator which flashes red when there is an error, such as request for an object that does not exist in the device. The yellow indicator flashes each time the VP4-2330 polls for master. The green LED to the right flashes each time a data request is replied to.

The yellow poll-for-master LED may flash quite frequently, or nearly never. How often the device polls for master depends on whether there is another master at some higher MAC address after the VP4-2330. If the next consecutive MAC address is in use by another device on the link, the poll for master will rarely flash. If the VP4-2330 has a low MAC address and there are few or no other devices with higher MAC addresses, the poll for master LED will be quite busy.

The token pass led (green LED on left) should always be flashing. If there is a small number of devices on the network, it will flash rapidly. The more devices there are on the network, the more slowly it will flash because there are more devices passing the token around. If it never flashes, there is a communication problem, such as baud rate mismatch.

The red LED is a communication error indicator most of the time. However, during restart it will be on solid. Immediately following startup, it may flash a fault code if a serious error has occurred.

## 2.7 Indicators on the MTX002 USB to MS/TP Adapter



The yellow LED next to the USB connector will light up any time the PC recognizes this device. There is a blue power LED inside the adapter visible by looking into the adapter through the slot next to the USB adapter. The power LED should always be on if connected to the PC. The yellow LED will only come on when recognized by PC software.

The LEDs next to the terminal block indicate network traffic on the MS/TP network.

The Token LED flashes green each time the adapter passes the token. The adapter functions as another MS/TP device on the network, and gets into the token passing loop just like any other MS/TP device.

The PFM LED flashes yellow each time the adapter sends a Poll For Master (PFM).

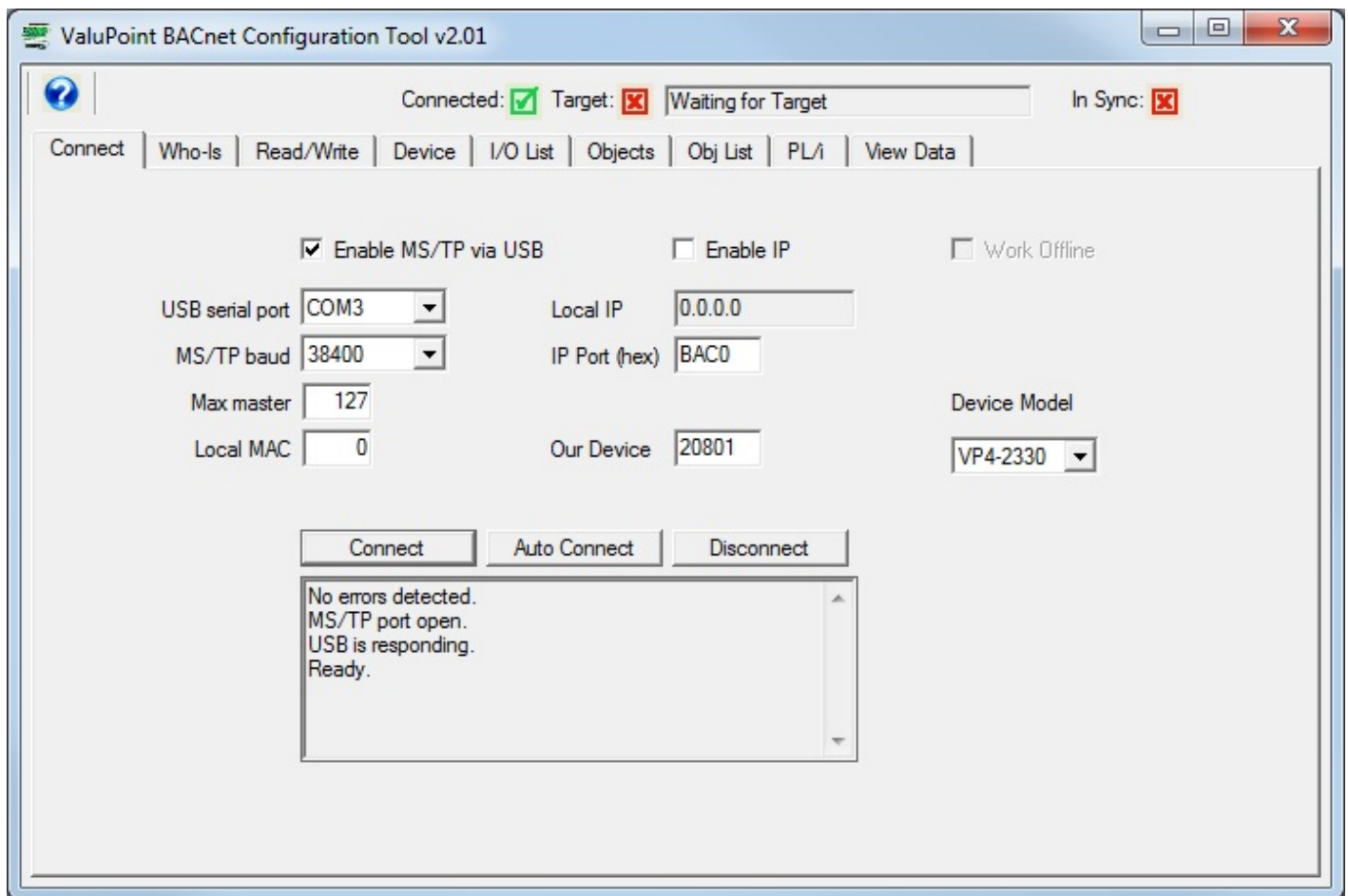
The Data LED flashes green each time a good packet other than token or PFM is received, or a packet other than token or PFM is sent, by the adapter.

The Error LED flashes each time a valid packet is received by the adapter, but resulted in an error. The Error LED can also indicate a timeout waiting for the slave device to respond (typically the device being configured).





## 3 Connect



### 3.0 Do This for First Time Startup

Assuming you have connected and installed the USB to MS/TP adapter (“dongle”), check the box ‘Enable MS/TP via USB’. Select the COM port assigned by your PC to your adapter. Find the Device Manager hardware properties in the control panel of your PC and look for the COM port if you have not done so already.

If connecting to a ValuPoint for the first time, as shipped from the factory, leave the remaining settings at their defaults.

Click the Connect button. You should see the message illustrated above indicating no errors, port open, USB is responding, and Ready. If not, stop here and determine why. Re-check the port number used. See that MTX002 shows up under Ports (COM & LPT)

in your PC's Device Manager hardware list. If MTX002 does not show up, revisit the USB installation above.

In addition to seeing "Ready", you should see the connected icon at the top change from a red X to a green check mark. When you see "Ready", you are ready to move on to the Who-Is page.

Note: You may see "No errors detected. MS/TP port open." even though your MTX002 is not connected. This means the configuration software found a valid COM port and opened that port, but there is no MTX002 on that port. If you do not see "USB is responding. Ready." then you are not ready.

**IMPORTANT:** Be sure to select the ValuPoint Device Model that you are about to configure. Having the wrong model selected will result in an incorrect configuration as well as errors when attempting to access properties that might not exist since the objects do not match up.

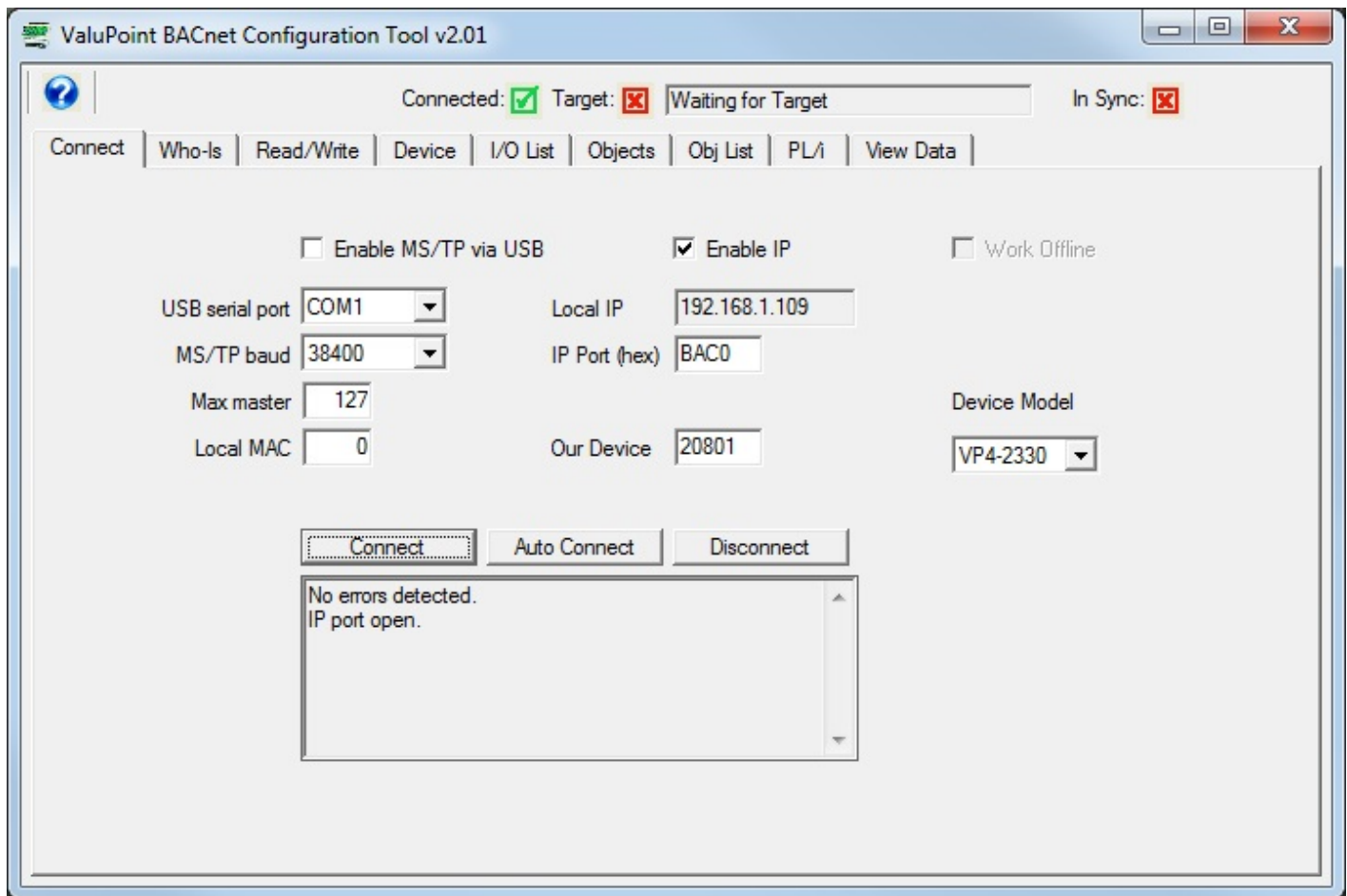
### **3.1 MS/TP**

Check the box 'Enable MS/TP via USB' if you will be accessing the ValuPoint via MS/TP (the most common approach). Select the COM port associated with your USB to MS/TP adapter. Select the baud rate your ValuPoint is configured to run (default is 38400). Set Max Masters and Local Mac to your known values if they have been changed. If device instance 20801 is already used by some other device, change the 'Our Device' number before connecting.

Note: The MS/TP interface for this software requires the MTX002. You cannot use a generic USB to RS485 adapter. This is because the MTX002 is an intelligent adapter that handles all of the token passing within the adapter, and only passes APDU's via USB.

### **3.2 IP**

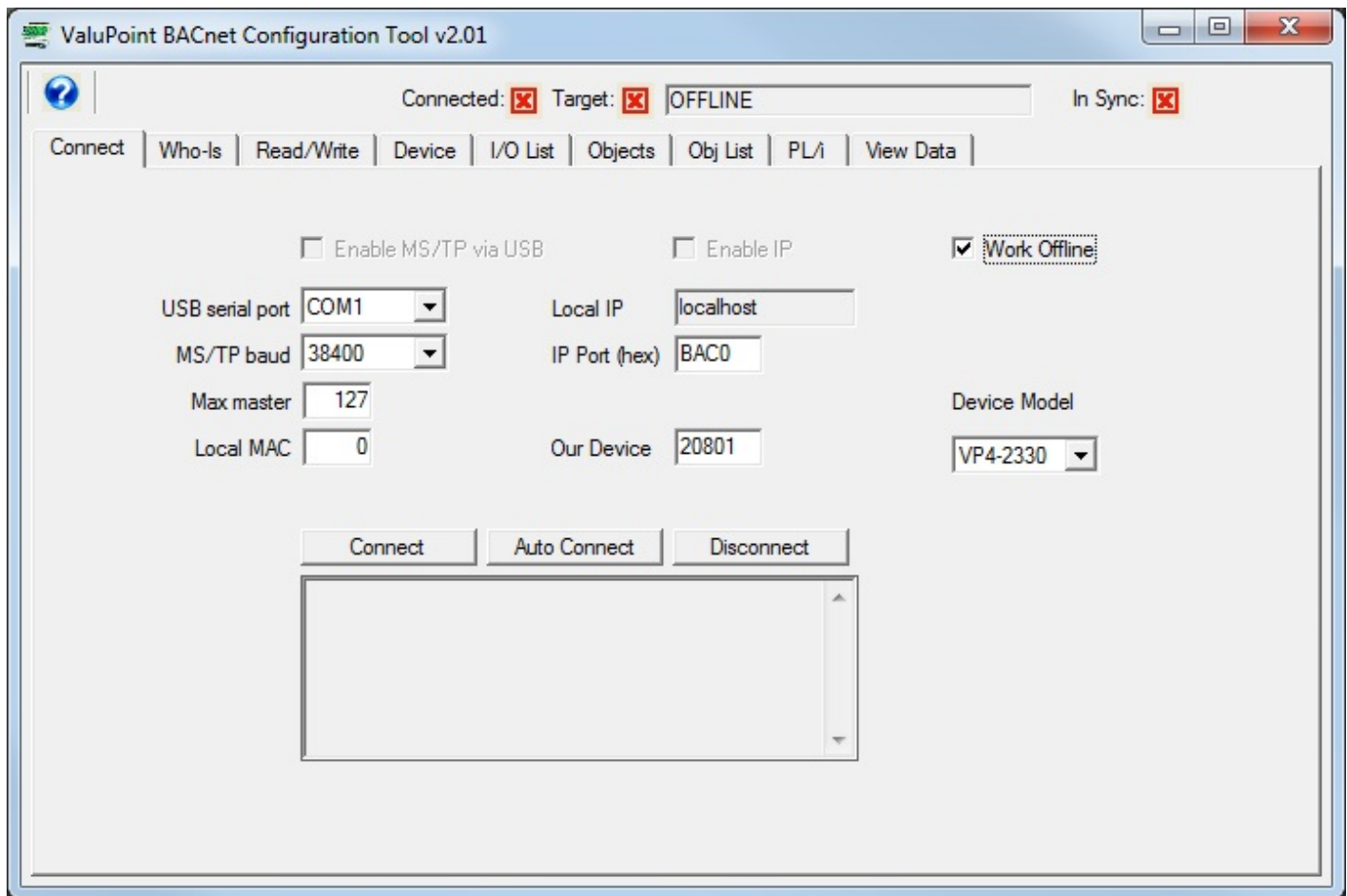
Check the 'Enable IP' box if you will be accessing the ValuPoint via IP, through a BACnet IP to MS/TP pass-through gateway or router. Enter the port number (0xBAC0 is the default) in hexadecimal. If device instance 20801 is already used by some other device, change the 'Our Device' number before connecting.



If the IP port is successfully connected, you will see the message "No errors detected. IP port open." In addition, you will see your PC's IP address appear in the Local IP window. You are now ready to move on to the Who-Is page.

### 3.3 Work Offline

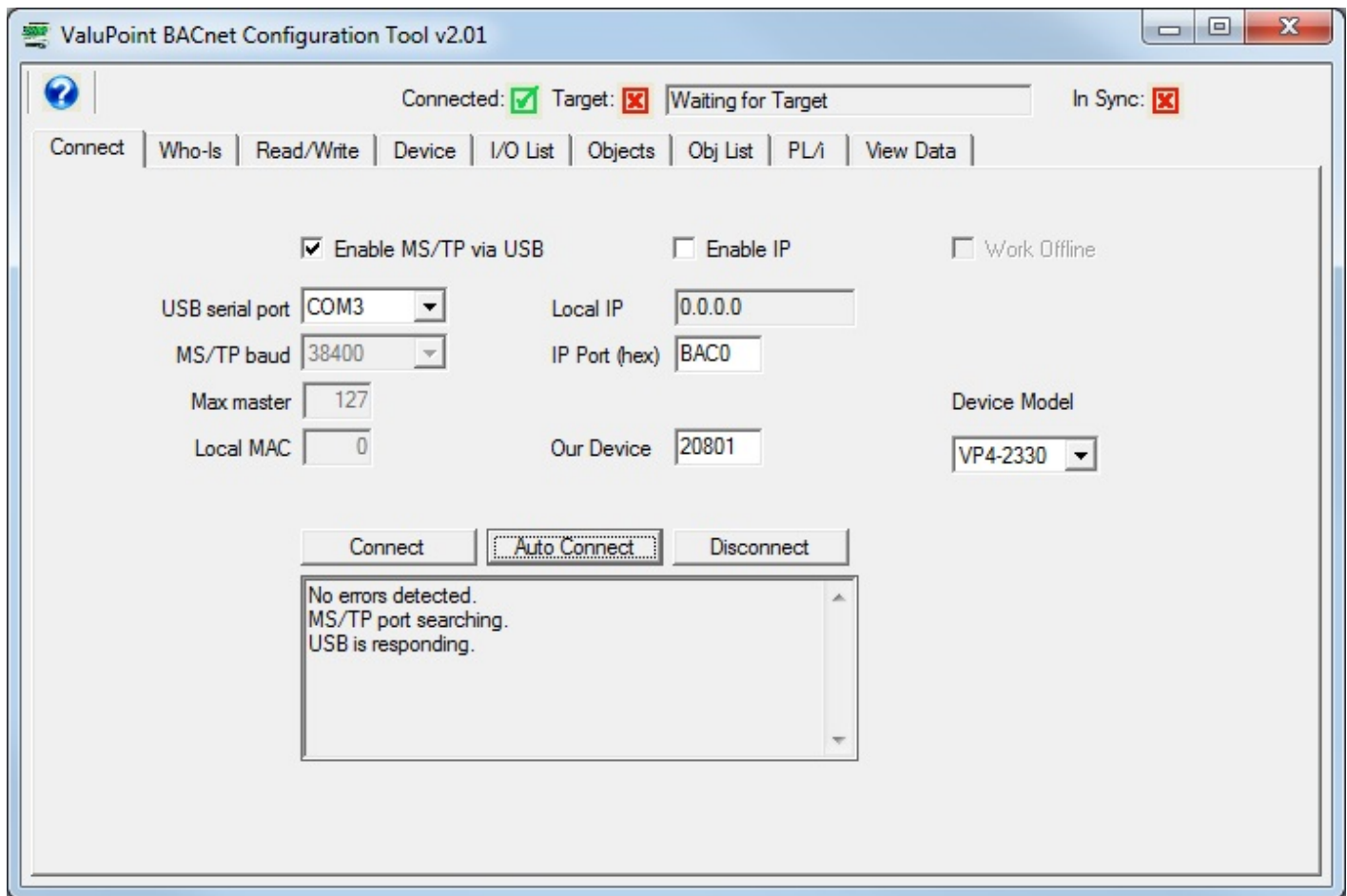
You will not be allowed to select most of the tabs in the tool until you have made the prerequisite device connection and selection. If you want to work offline, simply using the tool to create a configuration file that you will upload to a device later, check the 'Work Offline' box.



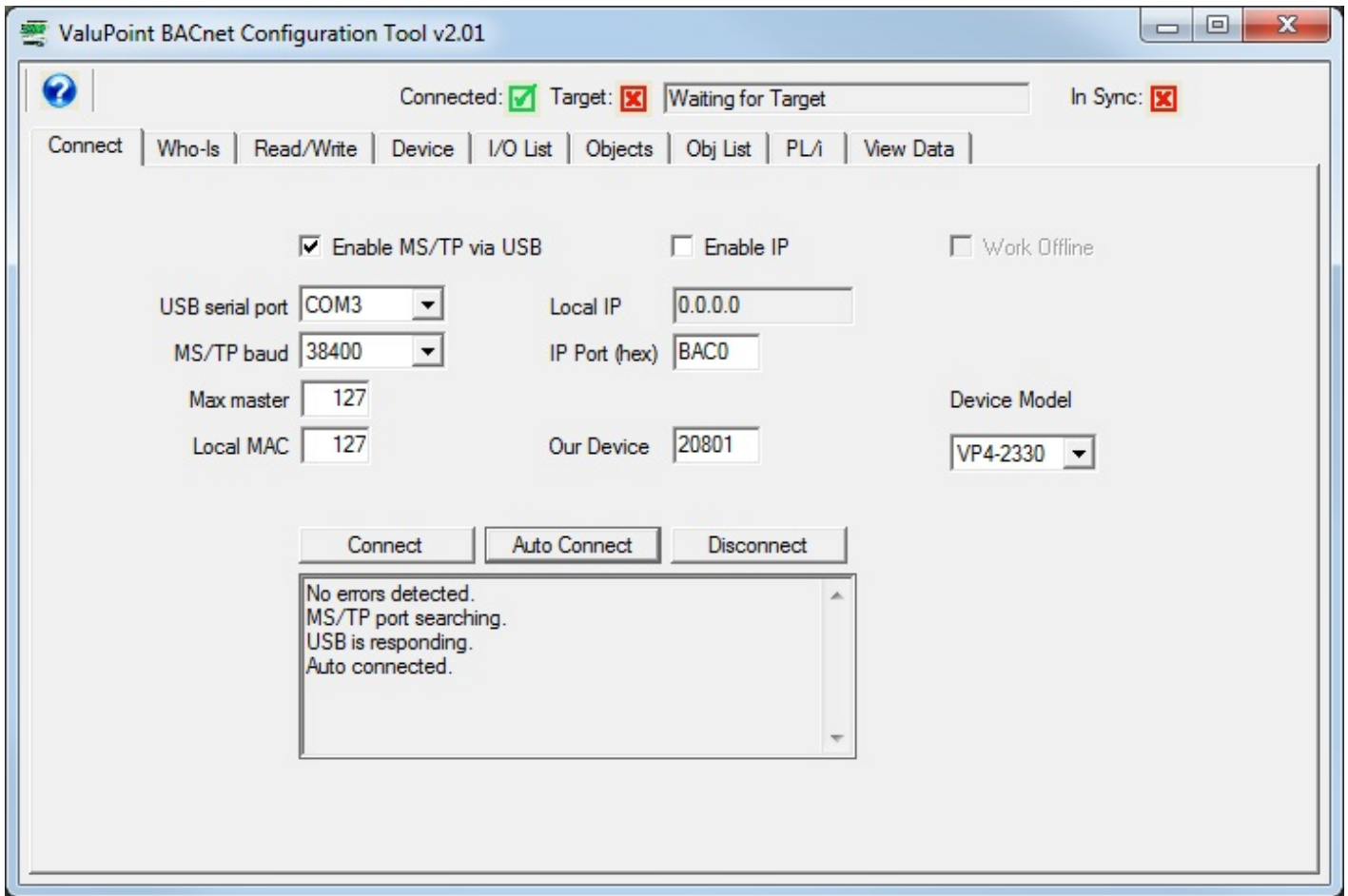
### 3.4 Auto Connect

The fastest way to connect is to select the known baud rate, max master setting and some unused local MAC. However, if you are connecting to a ValuPoint that has been previously configured and you do not know its settings, you can use Auto Connect to search for it. The Auto Connect does take some time because it is testing the various possible baud rates. Then when it finds a baud rate that seems right, it spends some time listening to polls for master to determine the max master setting that seems to be in effect. It is also looking for an unused MAC (station ID) to use for the tool itself. The tool will assign itself the highest numbered unused MAC address. Once a seemingly good set of parameters is found, these will be inserted in the appropriate windows for you, and you will see the indication 'Auto connected'. You may now proceed.

To auto connect, check Enable MS/TP via USB, then click the Auto Connect button. The screen will initially look like this:

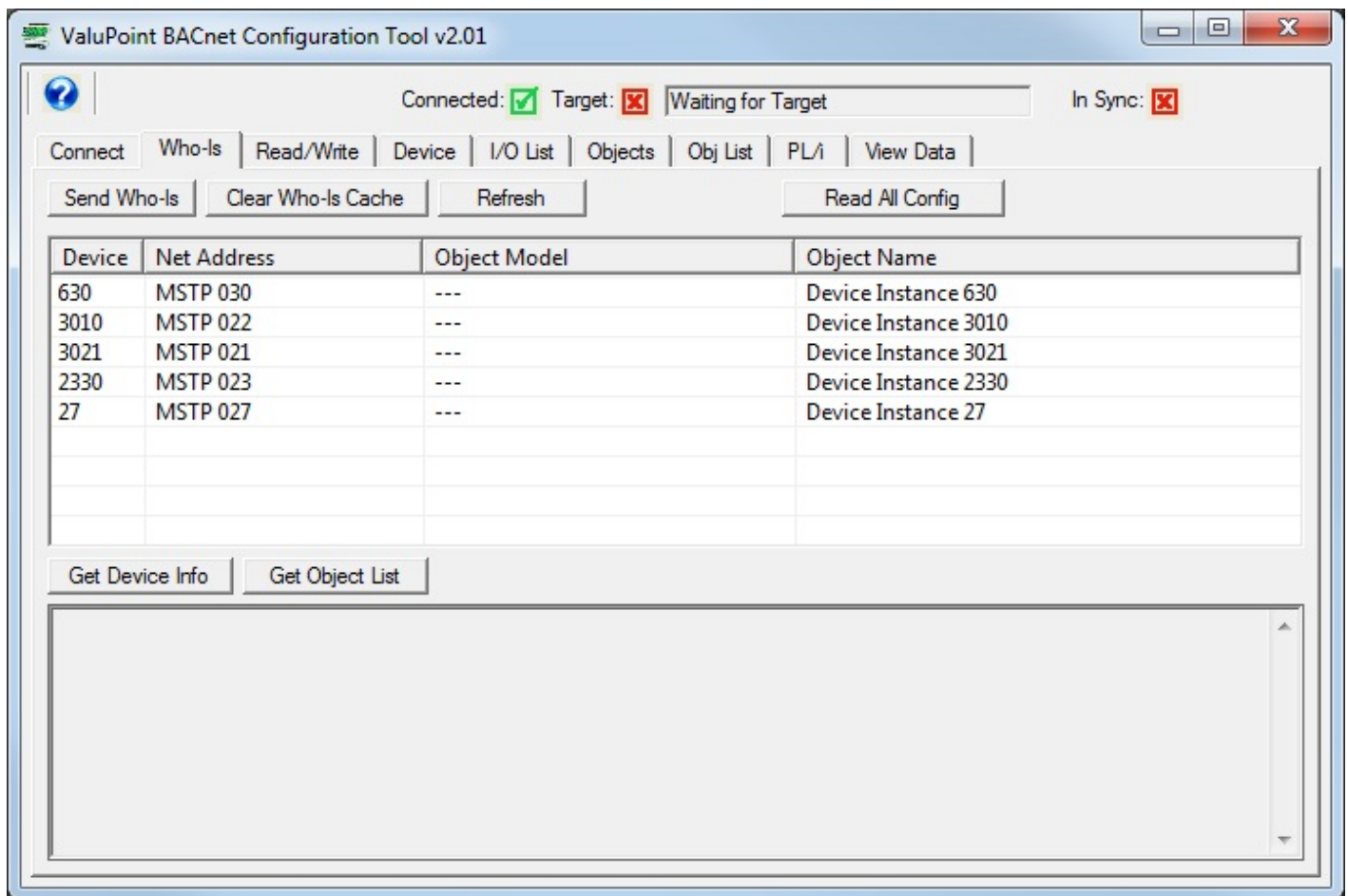


Once the auto connect process has completed, the disabled windows will become enabled, and will display the parameters found. When you see the message "Auto connected" show up below "USB is responding", you are ready to move on to the Who-Is window.





## 4 Who-Is



### 4.0 Do This for First Time Startup

Alternate between clicking 'Send Who-Is' and 'Refresh' until the device you wish to configure shows up on the list. Then double-click that device and observe that it shows up in the very top target identification window. The target icon at the top should change from a red X to a green check mark. Then, assuming the device is a ValuPoint, click the Read All Config button to get the configuration tool entirely in sync with the device.

### 4.1 Send Who-Is

The Who-Is page is used to locate and identify devices on the BACnet network, and

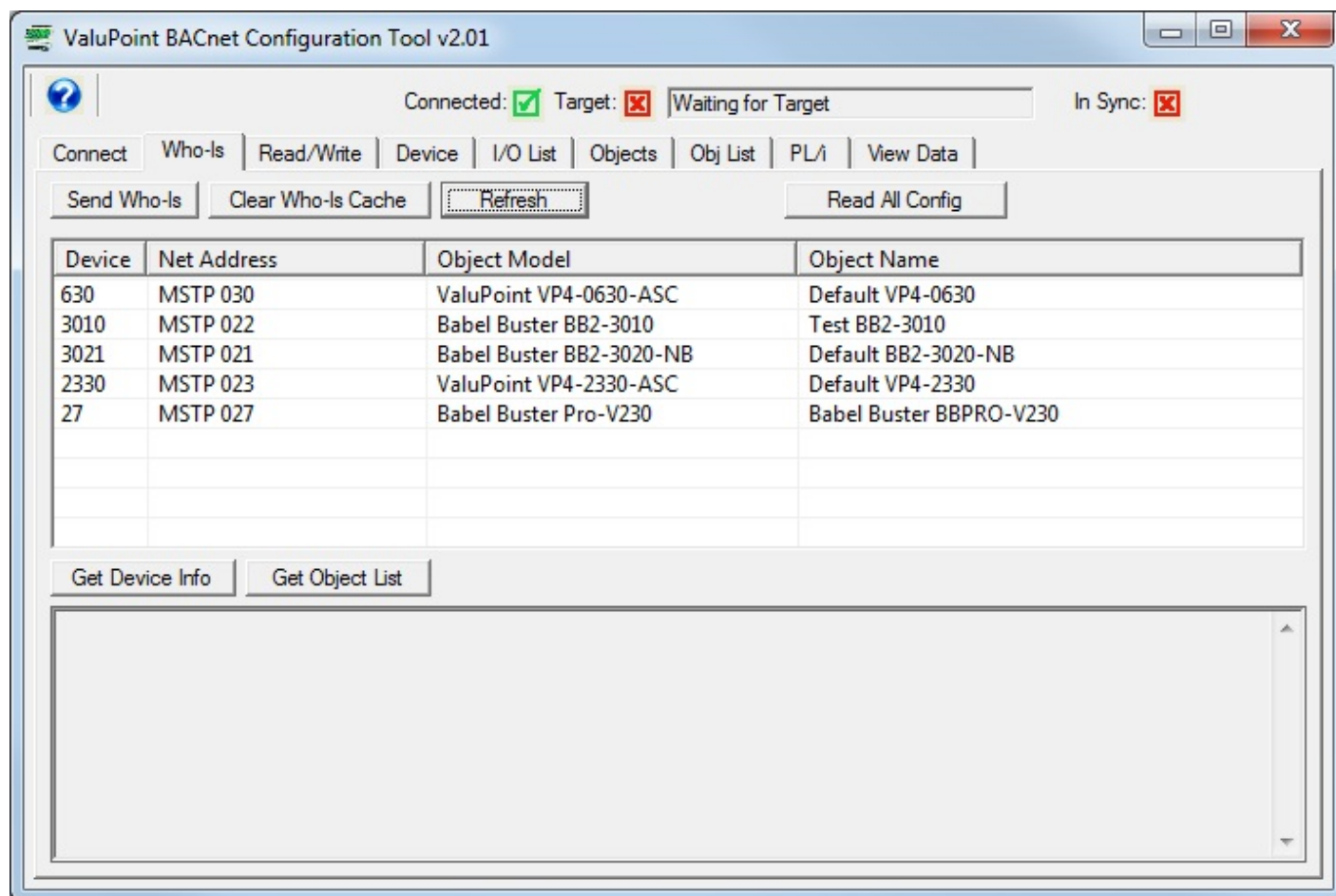
select the device that you wish to configure. By the time you switch to this tab after connecting (Connect tab), it is common to see some devices already listed. You can click 'Send Who-Is' any number of times to send out another request for devices to identify themselves.

The Who-Is only locates devices and lists them as illustrated in the first screen shot above. You need to use Refresh to get the object models and names.

The MAC address of MS/TP devices found will be listed in the Net Address column. IP devices on the other side of a router will also be shown, but at the same MAC address - this will be the MAC address of the router. Additional network information is recorded internally and the configuration tool will use this information as needed to access the devices on the other side of the router.

## 4.2 Refresh

Click 'Refresh' to see which devices have responded. Click 'Refresh' at least twice to cause the tool to request object model and name from each responding device. The 'Send Who-Is' sends out the who-is request, and 'Refresh' sends out read requests for model and name for any MAC that has not already been queried for this information. Following successful refresh, the screen will appear as illustrated below.

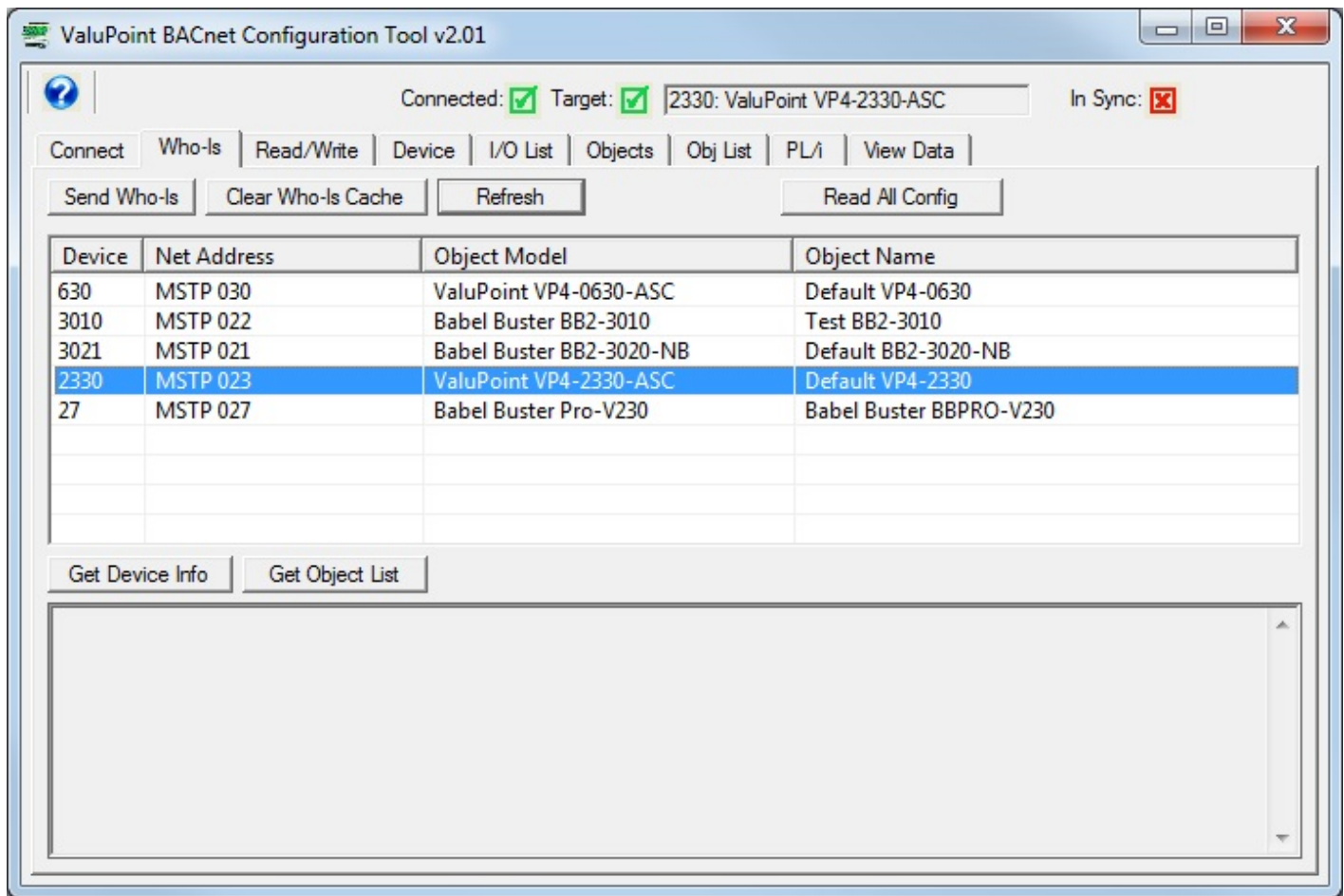


## 4.3 Select Target



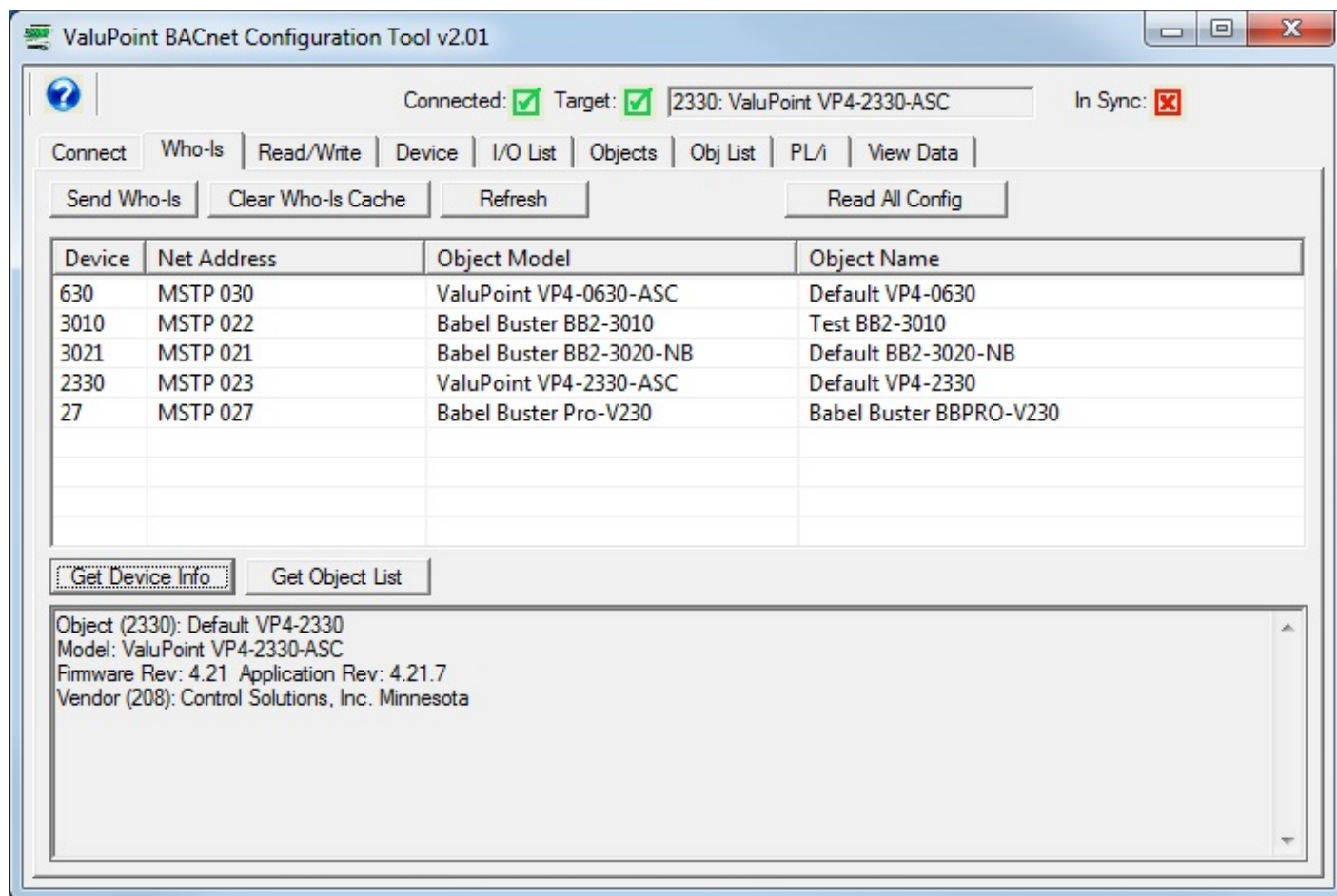
Double-click the device you wish to configure. Its device instance will appear in the target window at the top, along with the device name if it had been obtained by clicking 'Refresh' at least twice. The Target icon at the top should also change from a red X to a green check mark.

You must select a device before most of the other tabs will work. The only tab that will work without first selecting a device is the Read/Write tab which is generic in nature and does not actually do any configuration. If you attempt to access a feature that requires device selection first, you will get an error message to that effect.



#### 4.4 Get Device Info

Click the 'Get Device Info' button to get several pieces of information about the device as illustrated below. The Get Device Info button will work for any device selected as "target" regardless of whether it is a ValuPoint.



#### 4.5 Get Object List

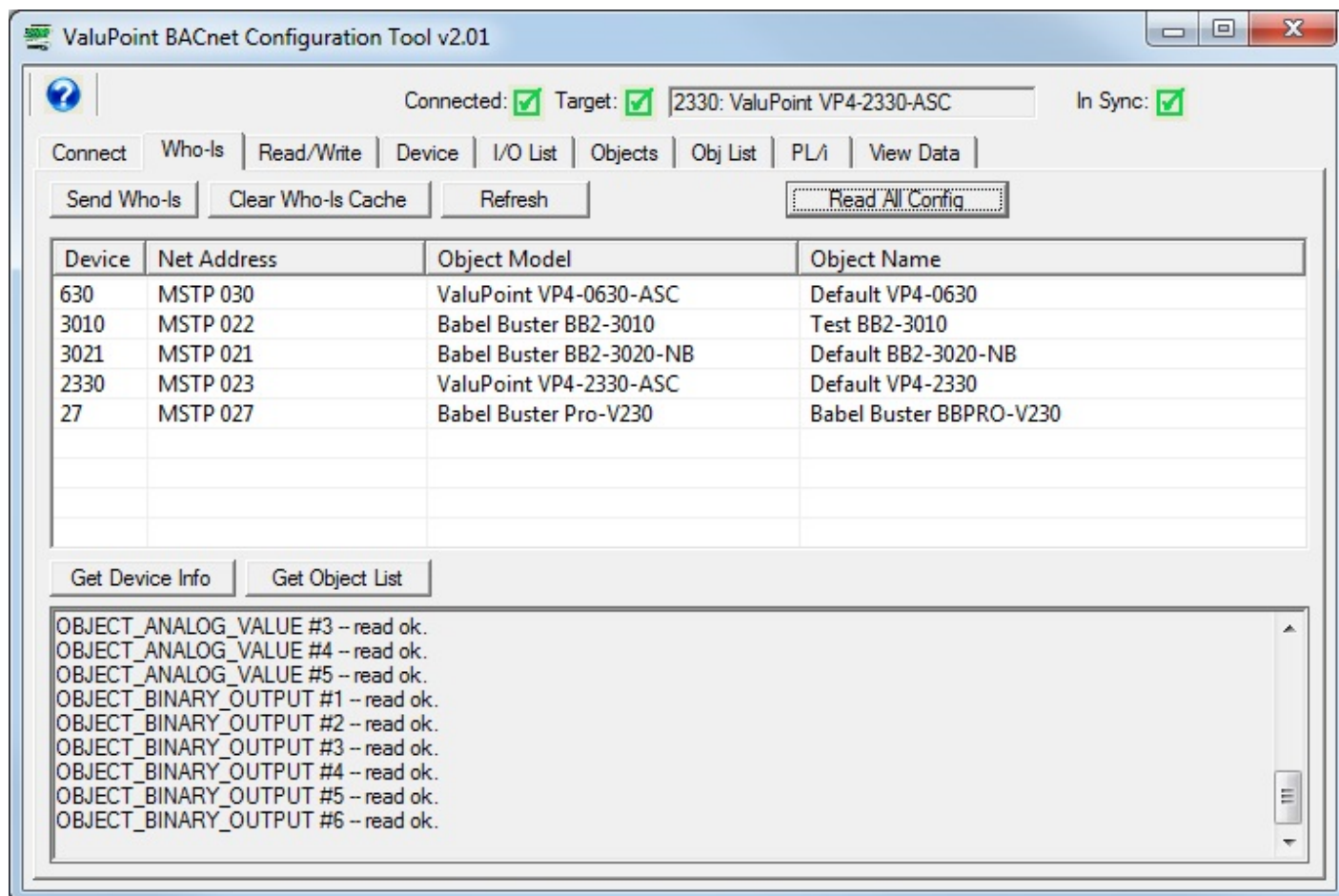
Click 'Get Object List' to get a list of all of the device's currently known objects. Note that the total object count contained in a ValuPoint is configurable. As shipped, the ValuPoint defaults to having one object per I/O point and no additional objects. Therefore if you were expecting to see the advertised 200 objects but only see a few, this is to be expected until you allocate additional objects.

#### 4.6 Clear Who-Is Cache

You normally have little use for this button. However, if you change the device's Object ID (device instance), you will need to return to this page, clear the cache, and repeat the device selection process because the device's identity has changed.

#### 4.7. Read All Config

There is a shortcut available for reading the properties found on various pages when getting a device in sync for the first time, usually after powering up the gateway. Click Read All Config to do the equivalent of reading properties on other pages sufficient to get all the red X's to change to green checkmarks.



The ValuPoint configuration tool will work when connected via a BACnet IP router (check Enable IP on Connect page). The Who-Is list of devices will appear as illustrated below when connected via IP. Note that multiple MS/TP devices will appear at the same IP address. This is the IP address of the BACnet IP router. Network number and MS/TP MAC address are recorded internally by the configuration tool and used as needed to access the MS/TP device via the router.

ValuPoint BACnet Configuration Tool v2.01

Connected:  Target:  545: ValuPoint VP4-2330-A In Sync:

Connect Who-Is Read/Write Device I/O List Objects Obj List PL/I View Data

Send Who-Is Clear Who-Is Cache Refresh Read All Config

Device	Net Address	Object Model	Object Name
733	IP 192.168.1.116:0xBAC0 ...	Babel Buster BB2-3010	BB2-3010 Test Device 3
545	IP 192.168.1.180:0xBAC0 ...	ValuPoint VP4-2330-A	Jims Test VP4-2330-A
711	IP 192.168.1.116:0xBAC0 ...	Babel Buster BB2-3010	BB2-3010 Test Unit 1
981	IP 192.168.1.178:0xBAC0 ...	Babel Buster BB2-3010	BB2-3010 on IP82
208031	IP 192.168.1.198:0xBAC0 ...	Babel Buster BB2-3010	BB2-3010 Gateway
176	IP 192.168.1.116:0xBAC0 ...	Babel Buster BB2-7030-01	Babel Buster BB2-7030 BACnet Router
722	IP 192.168.1.116:0xBAC0 ...	Babel Buster BB2-3010	BB2-3010 Test Unit 2
16001	IP 192.168.1.168:0xBAC0 ...	Babel Buster BB2-3010	Default BB2-3010

Get Device Info Get Object List



## 5 Read/Write

The screenshot shows the 'ValuPoint BACnet Configuration Tool v2.01' window. At the top, it indicates 'Connected: [checked]', 'Target: [checked]', and '2330: ValuPoint VP4-2330-ASC'. There is also an 'In Sync: [X]' indicator. Below this is a navigation bar with tabs: 'Connect', 'Who-Is', 'Read/Write', 'Device', 'I/O List', 'Objects', 'Obj List', 'PL/I', and 'View Data'. The 'Read/Write' tab is active. The main area contains several input fields: 'Object Type' (Analog Input), 'Object Instance' (1), 'Property' (Present Value), 'Array Index' (empty), 'Data Type' (Null), 'Priority' (None), and a 'Write Data' text box. There are 'Read Property' and 'Write Property' buttons. A log window at the bottom displays the value '0.006445'.

### 5.0 Skip This for First Time Startup

This page is not necessary for initially getting the device in sync with the tool. This page is also not necessary for performing any configuration functions.

This page may be used as soon as a device is selected on the Who-Is page, and the device does not have to be a ValuPoint to use the generic object property read/write on this page.

### 5.1 Read Property

Select object type, instance, property, and array index if any, and then click 'Read Property'. If the request is successful, the data will be displayed in the log window at

the bottom.

A collection of most often used property types are included in the drop list. If the desired property is not shown, select 'Other-->' and enter the property type code in the window next to the list. The property type codes are those defined by the BACnet standard. For example, Present Value can also be obtained using 'Other --> 85'.

## 5.2 Write Property

Select all of the same settings as you would for Read Property, and in addition, specify data type, priority (if commandable, use 'none' if not commandable), and data to write. Then click 'Write Property'.

Format for most data types is simply a numeric string. The data type 'character string' will consume everything found in the data window as ASCII text copied verbatim. Octet strings should consist of 8-bit values in hexadecimal notation (1 to 3 hex digits), with each octet separated by a comma. Bit strings should consist of a series of T and F characters, optionally separated by a comma. The first T or F is bit zero.

## 6 Device Page

ValuPoint BACnet Configuration Tool v2.01

Connected:  Target:  2330: ValuPoint VP4-2330-ASC In Sync:

Connect | Who-Is | Read/Write | Device | I/O List | Objects | Obj List | PL/I | View Data

Object ID (device instance)

Device Name

Device Description

Device Location

Auto-Reset Errors

Max Master

MS/TP Baud Rate

MS/TP Station/MAC Addr

New Password

Old Password

Object Allocation	Present	Pending	Unused
Basic Object Pool	<input type="text" value="200"/>	<input type="text" value="200"/>	<input type="text" value="200"/>
Commandable Object Pool	<input type="text" value="9"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Analog Input Count	<input type="text" value="14"/>	<input type="text" value="14"/>	<input type="text" value="0"/>
Analog Output Count	<input type="text" value="3"/>	<input type="text" value="3"/>	<input type="text" value="0"/>
Analog Value Count	<input type="text" value="5"/>	<input type="text" value="5"/>	<input type="text" value="0"/>
Binary Input Count	<input type="text" value="0"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>
Binary Output Count	<input type="text" value="6"/>	<input type="text" value="6"/>	<input type="text" value="0"/>
Binary Value Count	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Multistate Input Count	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Multistate Output Count	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Multistate Value Count	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

Device count read ok.

### 6.0 Do This for First Time Startup

Click 'Get Info' near left center of the screen. Then click 'Get Counts' near the lower right corner. This will read the device properties and get the tool in sync with the device.

Note: This has already been done if you clicked the 'Read All Config' button on the Who-Is page.

### 6.1 Get Info

Clicking the 'Get Info' button will read all of the information shown on the left half on this page, except that no password is ever read out (not permitted by the device).

### 6.2 Change Info

Changes to any of the device information on the left-hand half of this page may be made. Once changes are made, click the 'Change' button to register all changes except device instance (see Reassign). To cause changes such as baud rate, MAC address, or Max Master setting to take effect, it is necessary to reinitialize the device (see Reinit Device).

### 6.3 Reassign Device Instance

To change this device property, enter the new object ID (device instance) and click the 'Reassign' button. Once you do this, you will need to return to the Who-Is page, clear the cache, repeat the Send Who-Is and Refresh, and reselect your target device. You will get an 'unknown device' error if you try to continue without reselecting the device since the device ID has now changed.

#### 6.4 Get Counts

Click 'Get Counts' to read the present object allocation configured in the device. The counts will show up in the 'Present' column.

Working offline: Enter the object counts you will configure your device for at some point in the future, then click 'Get Counts' to accept the changes for working offline.

#### 6.5 Recalculate and Reconfigure Counts

Enter new object counts in the windows in the 'Pending' column. Click 'Recalculate' to check to see if the count is valid. The recalculate will also be automatic any time you move the mouse cursor from one window to another.

Begin by deciding how many commandable objects you want in the new configuration, and enter that number as the Commandable Object Pool. The Basic Object Pool will be recalculated to use the remaining resources. Commandable objects take more resources than basic objects. With zero commandable objects, you can expect to see 315 basic objects. The maximum number of commandable objects is 136, which leaves only 1 basic object. As you can see, there is more than a 2 to 1 ratio of resource requirements.

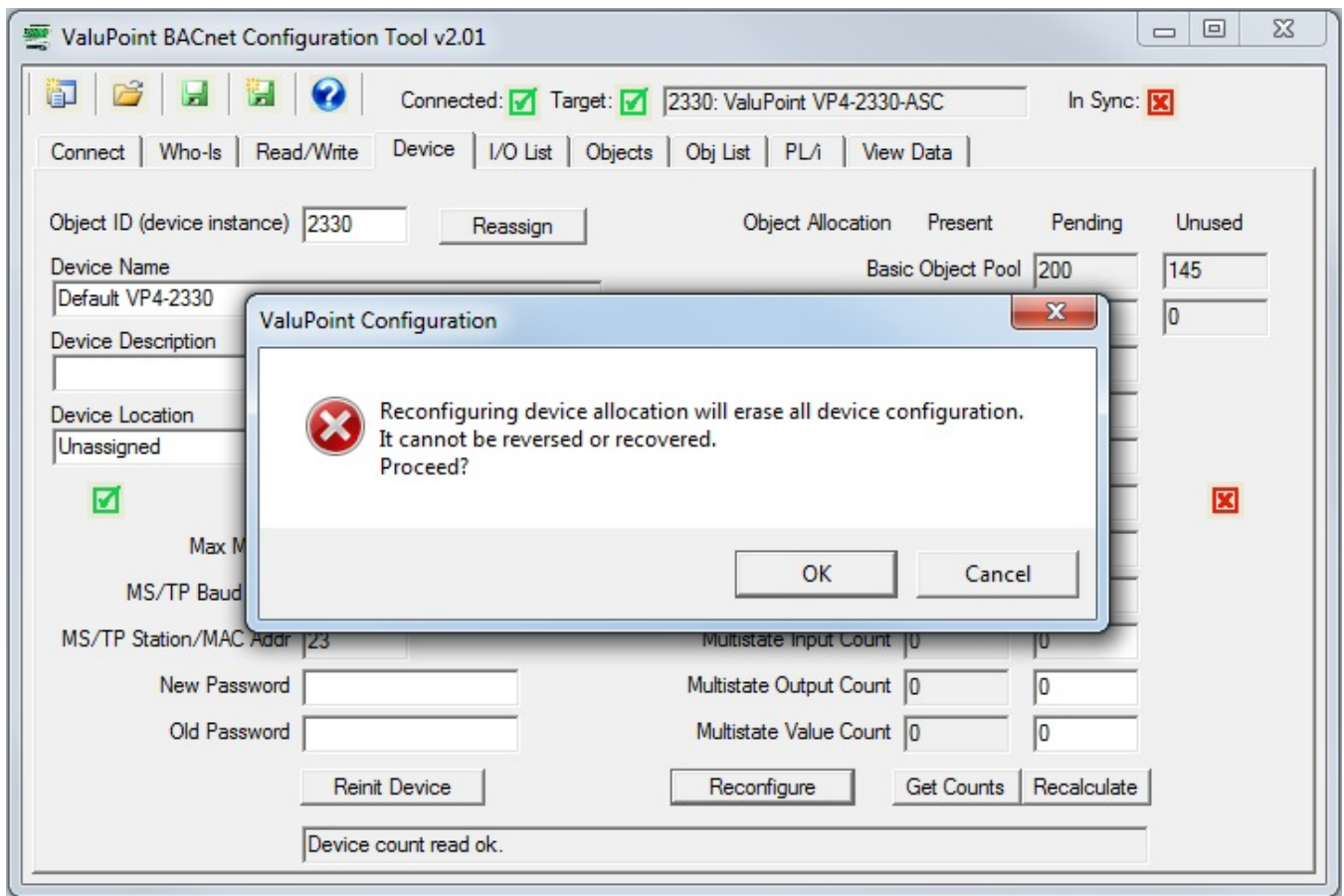
Once you have allocated some number of commandable objects in the pool, proceed to enter object counts for each of the object types. A count of zero is permissible.

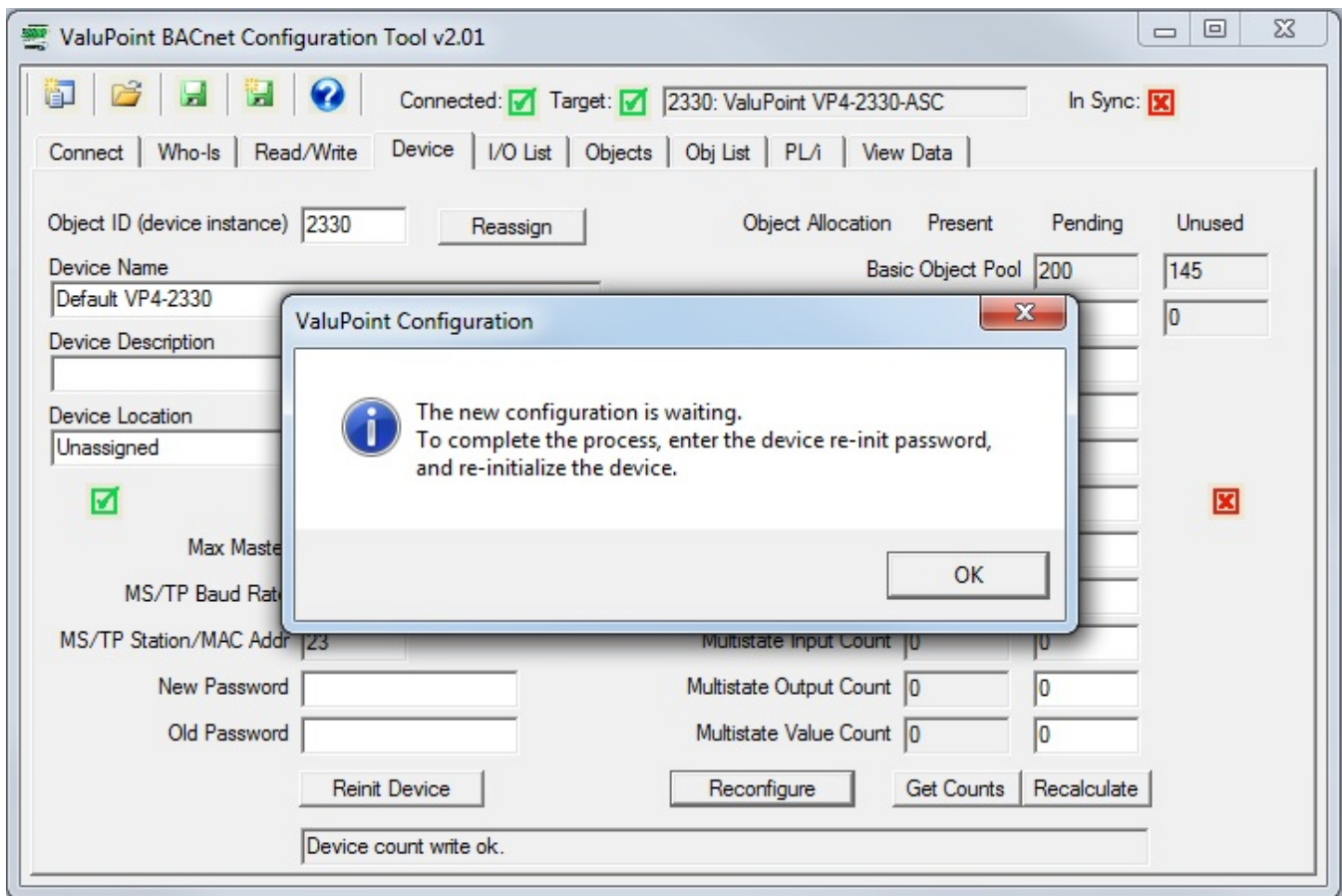
When you are ready to reconfigure the device with the new object allocation, click the 'Reconfigure' button. *NOTE: This action will completely erase all prior object map configuration.*

*If you get services error code 46, or an error message about invalid configuration, you are probably attempting to increase object counts toward the top of the list while decreasing counts further down the list in a device with a relatively full count. Start by reinitializing to all object counts set to zero except for Analog Inputs (which can be set to some low number). Now that the device sees available objects in the pool, you can proceed to allocate as desired.*

Upon clicking Reconfigure, you can expect to see the following sequence:







## 6.6 Reinit Device

Click 'Reinit Device' to restart the ValuPoint using new settings to baud rate, MAC address, or Max Master setting. You will need to enter the old password since the device reinitialize is password protected. If you had entered a new password when you clicked 'Change Info', it will become effective after using the old password one last time to cause the changes to take effect. NOTE: The default initial password is "valupoint".

If you have changed baud rate or other communications settings, you will most likely need to completely restart the configuration tool, starting with disconnecting and reconnecting on the Connect page.

## 6.7 File Read/Write (XML)

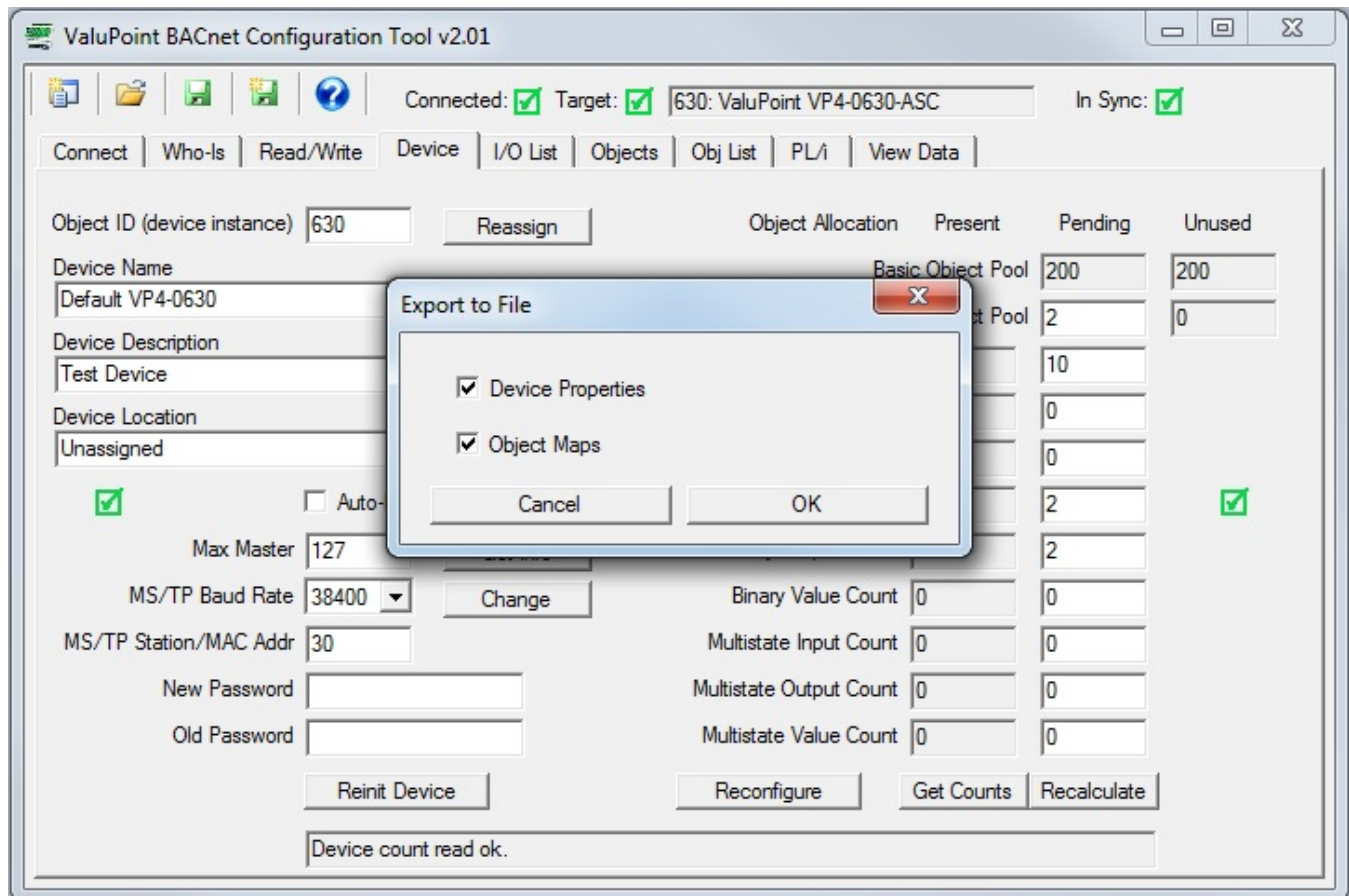
Click the 'new' icon to erase everything shown on the device page.

Click the 'open file' icon to read a device configuration from an XML file. If you will be replicating the same device configuration multiple times, you will save time by creating a file that can be reloaded. The currently open file will be indicated in the title bar of the program.

Click the 'save' icon to re-save the previously opened file, or to save for the first time if creating a new file. The presently displayed device configuration will be saved.

Click the 'save as' icon to create a copy of a previously opened file. A file dialog will appear allowing you to browse a selected directory and rename the file.

The file icons at the top of the tool screen will be different colors on different pages, and will be absent on some pages. These icons only appear where a file save is applicable. The different colors are meant to suggest different file types are used with different pages. There are 2 file types: (i) Device and Object Map XML, (ii) Object Map CSV. The XML file type will be saved from either the Device or Objects tabs while the CSV file type will be saved from the Obj List page.

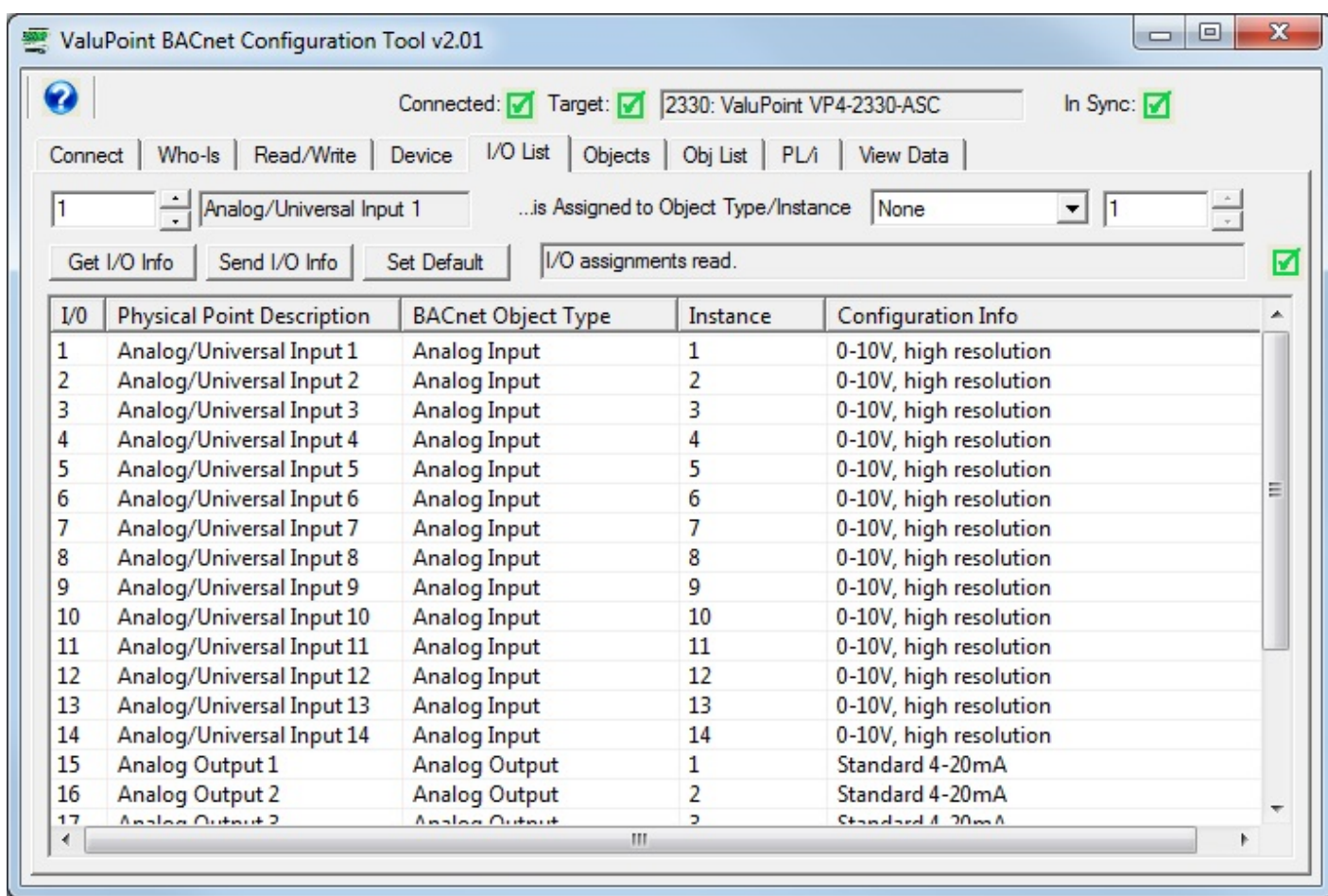


When opening or saving an XML file, the dialog illustrated above will appear allowing you to select device properties, object maps, or both. If you wish to replicate only the object mapping in another device without overwriting the other device properties, then you would select only the object maps.

The original ValuPoint configuration tool saved two different XML files, one for device properties and one for object maps. They have now been combined into one file. You may read the older individual files into the new tool. Simply select device properties or object maps when reading the old files.



## 7 I/O Point Assignments



### 7.0 Skip This for First Time Startup

When connecting and syncing for the first time, you should go to the 'Object Map List' page and read all object maps with one click of the mouse. The I/O assignment page will be filled in automatically when configuration is read from a device or loaded from a file.

If you do not wish to start with the default (or existing) hardware assignments, then you should first allocate object counts, then come to this page to assign I/O before moving on.

### 7.1 Assigning Physical I/O Points to BACnet Objects

The screen shot above illustrates the default I/O point assignment that the VP4-2330 is shipped with. It may not be necessary to make any changes. The most common change that will be made is to assign analog/universal inputs (A/UI) to Binary Input objects instead of Analog Input objects. The Binary Input objects would then be configured as discrete or dry contact inputs on the Objects page.

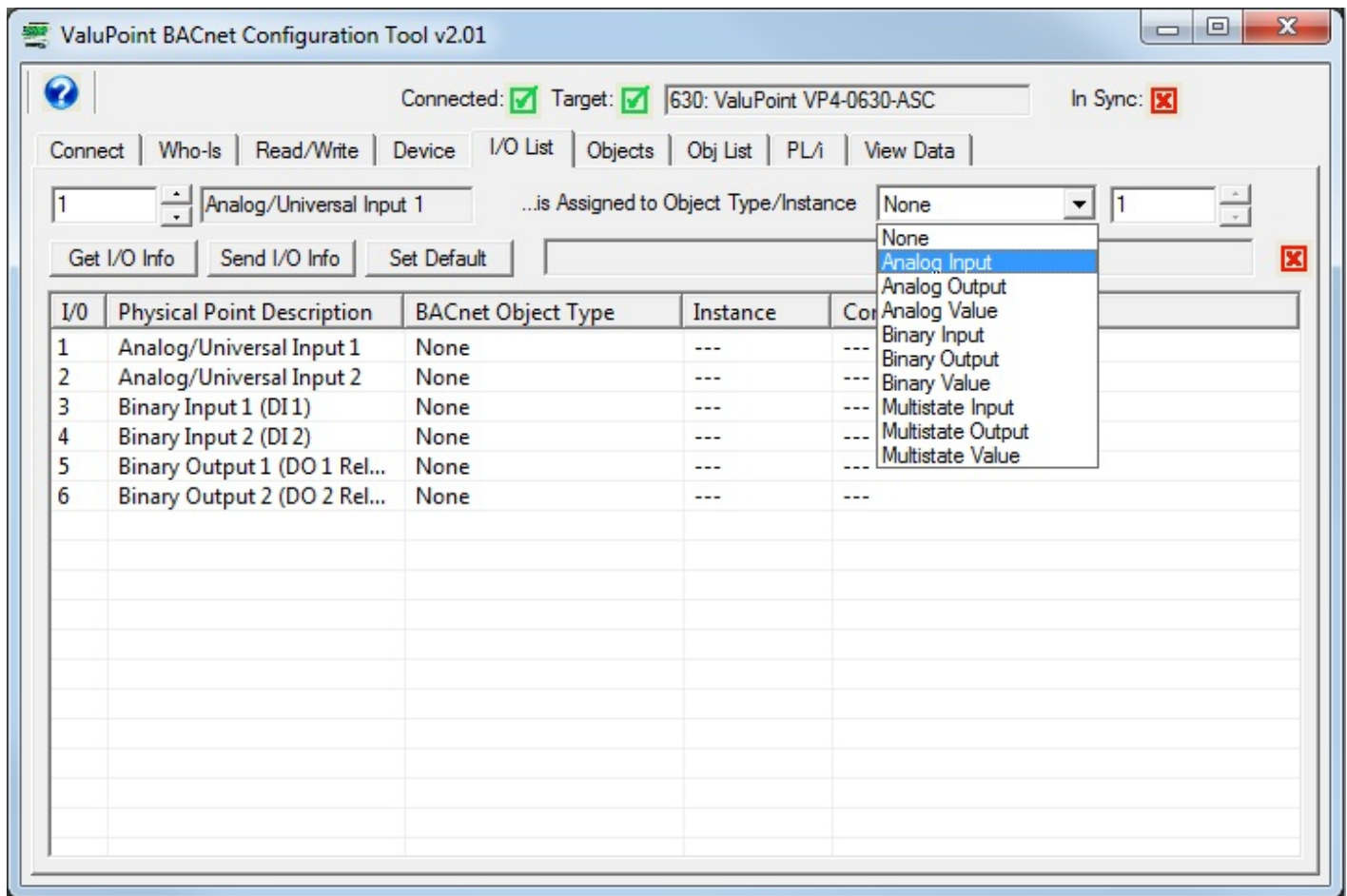
Assignments are made or changed using the top line of input windows. The large window is for display only. The steps for assigning an I/O point to a BACnet object are as follows:

- (i) Scroll the number on the left to the desired physical I/O point number. You can also double click on a line in the large display window to directly select that point.
- (ii) Select a BACnet object type from the drop-down list on the right.
- (iii) Scroll the number on the right to the BACnet object instance desired.

Get I/O Info – Click this button to read I/O assignments currently in effect in the device. Everything previously displayed on the page will be overwritten with information read from the device.

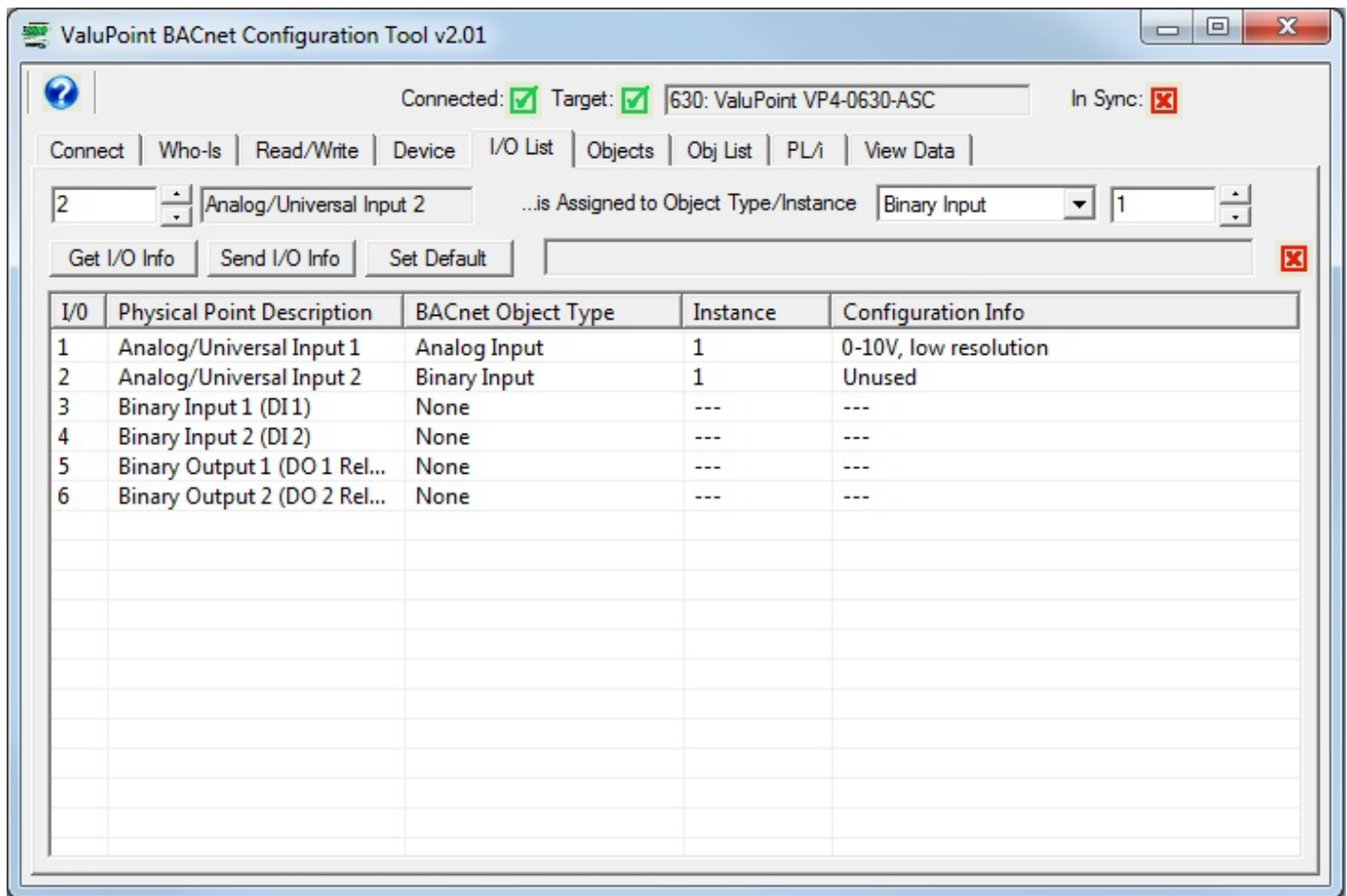
Send I/O Info – Click this button to send the I/O assignments displayed to the device. Until you click this button, the assignments only exist on your PC.

The screen shot below illustrates selecting physical I/O assignments for a VP3-0630 that has not been set to any default values.



You can only map an I/O point to an object that is not already mapped. If you see skips in object instances, it may be because some objects are already claimed by other points or other uses.

Configuration will initially be displayed as "Unused". Once I/O points have been associated with BACnet objects, proceed to the Objects page to configure the I/O point if you wish to make changes to the configuration of the selected I/O point.



The universal inputs default to analog objects, but can be assigned to binary objects. When a binary object is first selected, its configuration will be displayed as "unused". You may then go to the Objects page and configure the individual object to select how the hardware should process input for use as a binary input. The most common selection will be "dry contact, active closed".

The screenshot displays the ValuPoint BACnet Configuration Tool v2.01 interface. At the top, it shows connection status: "Connected:  Target:  630: ValuPoint VP4-0630-ASC In Sync: ". Below this is a menu bar with options: Connect, Who-Is, Read/Write, Device, I/O List, Objects, Obj List, PL/I, and View Data. The "I/O List" tab is active, showing a configuration for point 2: "Analog/Universal Input 2" assigned to "Binary Input" type, instance 1. Below the configuration fields are buttons for "Get I/O Info", "Send I/O Info", and "Set Default". The main area contains a table with the following data:

I/O	Physical Point Description	BACnet Object Type	Instance	Configuration Info
1	Analog/Universal Input 1	Analog Input	1	0-10V, low resolution
2	Analog/Universal Input 2	Binary Input	1	Dry contact, active closed
3	Binary Input 1 (DI1)	None	---	---
4	Binary Input 2 (DI2)	None	---	---
5	Binary Output 1 (DO 1 Rel...)	None	---	---
6	Binary Output 2 (DO 2 Rel...)	None	---	---





## 8 Object Map Page

### 8.0 Skip This for First Time Startup

When connecting and syncing for the first time, you should go to the 'Object Map List' page and read all object maps with one click of the mouse. You can read all properties here, but you must click once per object (as many as 200 times for a fully loaded device). Better yet, just click 'Read All Config' on the Who-Is page after selecting your target device.

### 8.1 Common Parameters

Object Type/Instance – Select BACnet object type from the list, and enter or scroll to the desired object instance. The object number will stop scrolling at the limit set by

object counts on the Device page.

Object Name – Enter an alphanumeric name for the object. This name must be unique within the device.

Description – Enter an alphanumeric description for the object. This is informational only.

Units – Select a BACnet units type from the list.

Read Periodic (check box) – Check this box if the BACnet object will be periodically reading from the slave Modbus device or slave BACnet device. 'Read' is valid for Input and Value type objects.

Write Periodic (check box) – Check this box if the BACnet object will be periodically writing to the slave Modbus device or slave BACnet device. 'Write' is valid for Output and Value type objects.

Write on Delta (check box) – Check this box if the BACnet object will be writing to the slave device only when the object's value changes by a specified 'Delta'. It is valid to check both Write Periodic and Write on Delta at the same time, or check just one or the other. 'Write' is valid for Output and Value type objects.

Set Default on Power-Up (check box) – Check this box if the object should assume the default value every time the ValuPoint powers up.

Set Default on Comm Fail (check box) – Check this box if the object should assume the default value when communication with the slave device has failed some number of times.

Enable Max Quiet Time (check box) – Check this box to enable maximum quiet time. This is applicable to an object which is set to Write on Delta. The result is that if there has been no change within the max quiet time, the slave device will be re-written anyway.

Object is Persistent (check box) – Checking this box means the Present Value of the object will be retained through restart or power cycle, rather than reset to zero. This is applicable to only certain objects. Analog Input objects mapped to physical inputs, for example, will not honor the Persistent flag - an Analog Input will always reflect the actual input value. Analog Outputs will also not honor the Persistent flag - if a specific power-up value is desired, use the Default Value and check 'Set Default on Power-Up'. Analog Value objects will always honor the Persistent flag. Binary and Multistate objects follow similar rules.

Poll Rate (Sec) – This rate applies to periodic reading or writing, and simply specifies how often the slave device is read from or written to. (This parameter does not apply to physical I/O.)

Initial COV Increment – The initial COV increment specified here will apply when

another BACnet client subscribes to this object's COV with a simple Subscribe COV. This initial increment will remain in effect until some other BACnet client writes a new COV increment to this object. If the BACnet client does a Subscribe COV Property, then the COV increment specified in the subscription is used instead of this Initial COV Increment. COV Increment applies only to Analog objects. COV will be reported on any change in a Binary or Multi-state object.

Initial Relinquish Default – The Initial Relinquish Default will take effect when the ValuPoint is first powered up. After that, any BACnet client can write a new Relinquish Default value to the object. The Relinquish Default value applies to any Commandable object, and the Present Value of the object will assume this value when all 16 priorities have been relinquished.

Slope/Scale Factor – Scaling applies the formula  $y=mx+b$ . When reading from the slave, the raw data as read is multiplied by the scale factor, then the offset is added to produce the resulting Present Value. When writing to the slave, the offset is first subtracted from Present Value, and that result is divided by the scale factor to produce the raw data actually written to the slave device. NOTE: If no scale factor is given (zero is entered), no scaling will be done, as if slope=1 and intercept=0.

Intercept/Offset – The offset portion of the scaling as noted above.

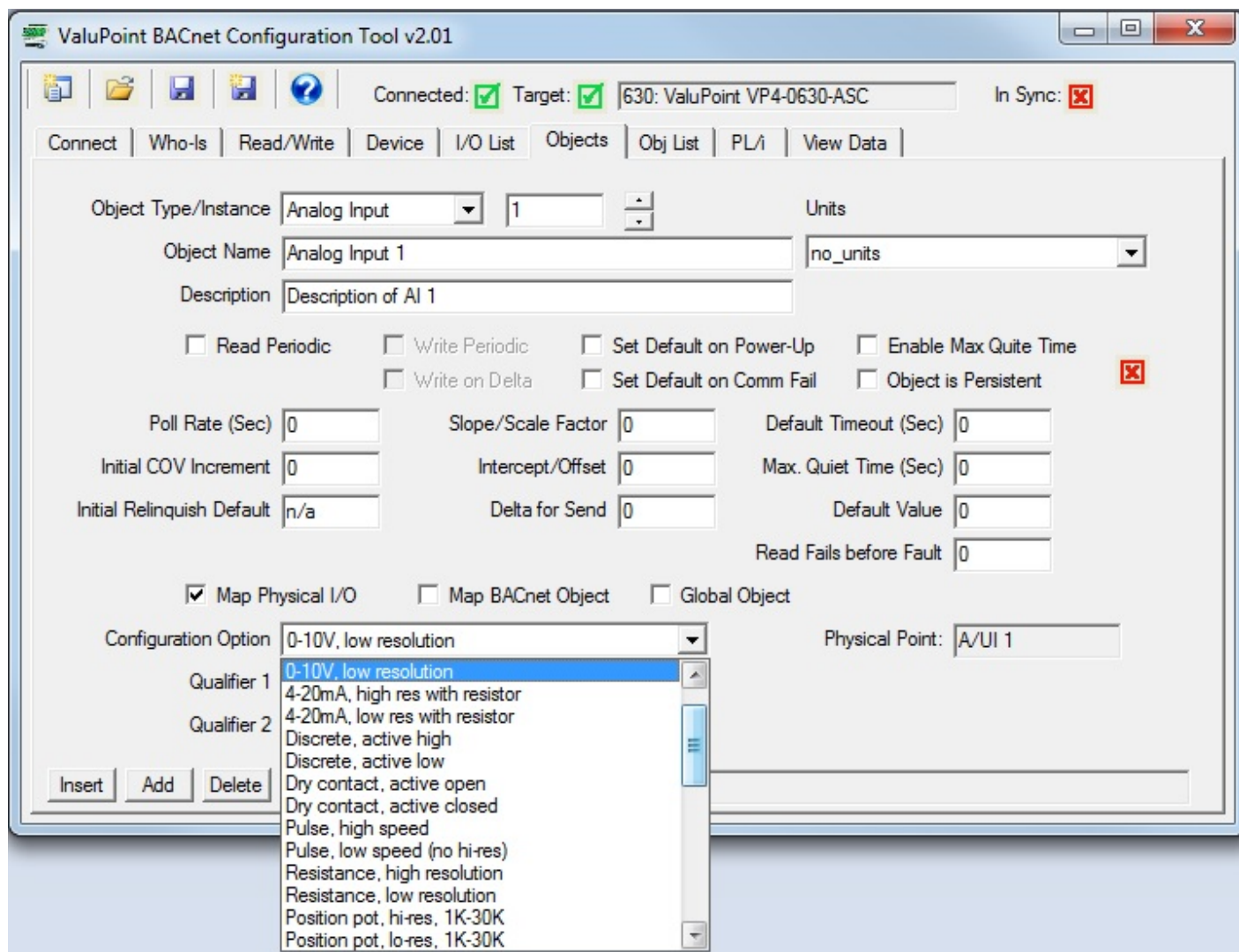
Delta for Send – Enter the threshold for writing when configured to Write on Delta. If a value of 5.0 is entered, the value must change by more than 5.0 before the slave device will be written to. Delta applies only to Analog objects. Binary and Multi-state objects will send on any change when Write on Delta is enabled.

Default Timeout (Sec) – Timeout for Modbus slaves is specified on the Modbus page. When the slave device is another BACnet device, the timeout given here specifies the amount of time the ValuPoint will wait for a response before calling it an error.

Max. Quiet Time (Sec) – This is applicable to an object which is set to Write on Delta. The result is that if there has been no change within this amount of time, the slave device will be re-written anyway. In addition to entering a nonzero value here, you must check the 'Enable Max Quiet Time' box.

Default Value – This is the value that should become the Present Value upon power-up or upon communications failure, if either of these options are selected by the appropriate check boxes above.

Read Fails before Fault – When 'Set Default on Comm Fail' is checked, this entry will allow you to disregard a small number of spurious errors. At least this many errors must occur consecutively before a fault will actually be flagged. Causing a fault as a result of a single instance of spurious noise on a communication line can be a nuisance. This setting allows quieting the nuisance fault notifications. (This parameter does not apply to physical I/O.)



## 8.2 Physical Hardware I/O Parameters

**Map Physical I/O** (check box) – Check this box to map the displayed BACnet object to a physical I/O point. The I/O point that has been set to be associated with this object will appear in the Physical Point window to the right. To change the association between I/O points and BACnet objects, go to the I/O List page and make changes there. If an object has already been mapped by checking this box, be sure to uncheck it first to release the object for reassignment before making changes on the I/O List page.

**Configuration Option** – Select the I/O configuration desired for the physical point. This only applies to A/UI analog/universal input points. The A/UI inputs may function as any of the following:

- 0-10V, high resolution
- 0-10V, low resolution
- 4-20mA, high res with resistor
- 4-20mA, low res with resistor
- Discrete, active high

Discrete, active low  
Dry contact, active open  
Dry contact, active closed  
Pulse, high speed  
Pulse, low speed (no hi-res)  
Resistance, high resolution  
Resistance, low resolution  
Position pot, hi-res, 1K-30K  
Position pot, lo-res, 1K-30K

Thermistor, 10K type III hi-res, F  
Thermistor, 10K type II hi-res, F  
Thermistor, 3K type II hi-res, F  
Thermistor, 20K type IV hi-res, F  
Thermistor, 5K type II hi-res, F  
Thermistor, 10K type III hi-res, C  
Thermistor, 10K type II hi-res, C  
Thermistor, 3K type II hi-res, C  
Thermistor, 20K type IV hi-res, C  
Thermistor, 5K type II hi-res, C

Thermistor, 10K type III lo-res, F  
Thermistor, 10K type II lo-res, F  
Thermistor, 3K type II lo-res, F  
Thermistor, 20K type IV lo-res, F  
Thermistor, 5K type II lo-res, F  
Thermistor, 10K type III lo-res, C  
Thermistor, 10K type II lo-res, C  
Thermistor, 3K type II lo-res, C  
Thermistor, 20K type IV lo-res, C  
Thermistor, 5K type II lo-res, C

The designation hi-res and lo-res refer to the A/D resolution. There are two types of A/D in the ValuPoint. The 'hi-res' A/D is a delta-sigma integrating converter that produces 15-bit results (16-bit including sign, which is always positive with the input circuitry implemented). The 'lo-res' A/D is a fast successive approximation converter which produces 10-bit results.

Due to the fact that 'hi-res' conversion is relatively slow (about 1/4 second compared to a few microseconds for lo-res), some combinations of I/O are not possible. Low speed pulse counting to 10Hz is only possible when all inputs are in lo-res mode. All discrete or dry contact modes are lo-res. High speed pulse counting (to 1kHz) is possible with any combination of modes; however high speed counting uses dedicated hardware and is only available in inputs channels 13 and 14.

Low speed pulse counting in combination with hi-res input conversions is possible, but input pulse rate is limited to about 2Hz. To increase this limit to 10Hz, all analog

inputs must be using lo-res modes.

Qualifier 1 – Enter the configuration qualifier value, if applicable, for the selected configuration. Qualifiers are required only in the following modes:

*4-20mA modes:* The qualifier is the resistance in ohms of the dropping resistor used to convert the current to voltage. An external resistor must be provided, connected between the A/UI input and ground/common. The resistor needs to be 1/2 watt (2 watt to withstand 24V power), and is left external simply because miswiring the 4-20mA sensor can easily apply 24V power directly to the input and cause the dropping resistor to heat up and possibly fail. The external resistor is simple to replace, whereas an internal resistor on a circuit board would be more trouble to replace.

*Discrete and Dry Contact modes:* The qualifier is a threshold between 1% and 99% at which the input should trip from off to on or vice versa. Although the A/UI inputs are specified as 0-10V inputs, they will actually measure up to 11.4V accurately in hi-res mode, and up to 15.0V in lo-res mode. Therefore, since discrete inputs are sampled as lo-res analog values and compared to a threshold, the qualifier here is a percentage of 15V for the trip point. A value of 50% will mean a threshold of around 7.5V.

*Pulse, Low Speed mode:* Because low speed pulse input is really a different interpretation of discrete input, a threshold setting is required for pulse sensing. The same threshold as used for discrete input applies here. Without any threshold (or with value of zero), no pulse counting will occur.

*Position Pot:* Position is simply a different interpretation of resistance measurement. The value resulting from position pot measurement will be a percentage from 0% to 100%, but this percentage will be a ratio based on the resistance value in ohms provided as the qualifier.

Qualifier 2 – Enter a second configuration qualifier value if instructed to do so. This qualifier is reserved for future use.

Physical Point – The physical I/O point assigned to the displayed object is indicated here.

### 8.3 MS/TP BACnet Client Parameters

Map BACnet Object (check box) – Check this box when the ValuPoint should look for another BACnet device as the slave device (instead of a Modbus slave). This box can be checked any time, but is of particular interest when using the ValuPoint as a Modbus slave and BACnet master for connecting BACnet devices to a Modbus network.

Remote Device Instance – Enter the device instance of the remote BACnet slave. The remote device must be able to support Who-Is/I-Am for locating device by device instance.

Object Type – Select the object type from the list, such as Analog Input, etc.

Mac address (no who-is) – Sets the MS/TP Mac address (station ID) if the slave device does not support Who-Is and I-Am. Device instance will be used to automatically capture the Mac address when Who-Is/I-Am is supported. This setting should be zero to use Who-Is, or set to some non-zero Mac address if the slave does not support Who-Is.

Property – Select the property to be read/written. A collection of most often used property types are included in the drop list. If the desired property is not shown, select 'Other-->' and enter the property type code in the window next to the list. The property type codes are those defined by the BACnet standard. For example, Present

Value can also be obtained using 'Other --> 85'.

**Instance** – Select the object instance. There are usually multiple instances of objects in a device. This identifies which one you want.

**Array Index** – Some properties are an array of values. If applicable, enter the array index here. Enter -1 to indicate no array index is to be used. Zero is a valid index.

**Bit Num** – When the property type is a bit string, this entry may be used to specify which bit you wish to single out, especially when applying the result to a Binary object. If no bit number is provided, the entire bit string will be assembled as an integer value representing a mask containing all bits.

**Data Type** – Select the data type expected from this list. Note that some types are not supported here. Character string, for example, is one type of data that cannot be forced into Analog, Binary, or Multi-state objects by the ValuPoint.

**Priority** – You must use a priority level when writing to a Commandable object. If the object is not commandable, you must select 'None' here to avoid having the request rejected.

## 8.4 Global Object

You may wish to set aside an object that is simply there to hold data written by



another BACnet device, or generated by the internal PL/i program. You would not want this object mapped to physical I/O or to some property in another BACnet device. The means of setting aside or 'reserving' an object is to simply check 'Global Object' instead of either of the Map options. If none of the Map or Global check boxes are selected, the object will not be included in the View Data list or in the XML file defining the objects.

## 8.5 Insert, Add, Delete

Clicking 'Insert' will insert a blank map before the currently displayed object map. Clicking 'Add' will insert a blank map after the currently displayed object map. Clicking 'Delete' will remove the currently displayed map and slide the remainder of the list up by one slot on the list.

The only time position matters much in the list is when configuring a packed register that maps a single Modbus register to multiple BACnet objects. The members of the packed register must appear in consecutive maps unless the 'Pack Mixed Object Types' is checked. Do not check 'mixed object types' unless you really need to mix analog and binary objects (for example) in the same packed Modbus register. Packing mixed object types is very inefficient since the gateway must scan the entire list of object maps for every instance of packed register. When packing the same object types, placing them in consecutive maps results in very efficient execution.

## 8.6 Read/Write Device

When changes are made, the in-sync icon changes to a red X on the right hand side of this page. When properties are either read or written, the device is now in sync, and the red X changes to a green check mark. This icon will change on an object by object basis indicating those objects that are in sync.

When all aspects of device configuration are in sync, between device and tool, the 'In Sync' icon in the top right corner of the screen will change to a green check box. This is the master In-Sync icon, and will be green only when all other in-sync icons are also green. This icon is effectively a summary icon.

**IMPORTANT:** Configuration changes have only made it as far as your PC if you see the red X on this page. If your hardware is not behaving as expected and you see a red X on this page, the hardware is probably not configured.

## 8.7 File Read/Write (XML)

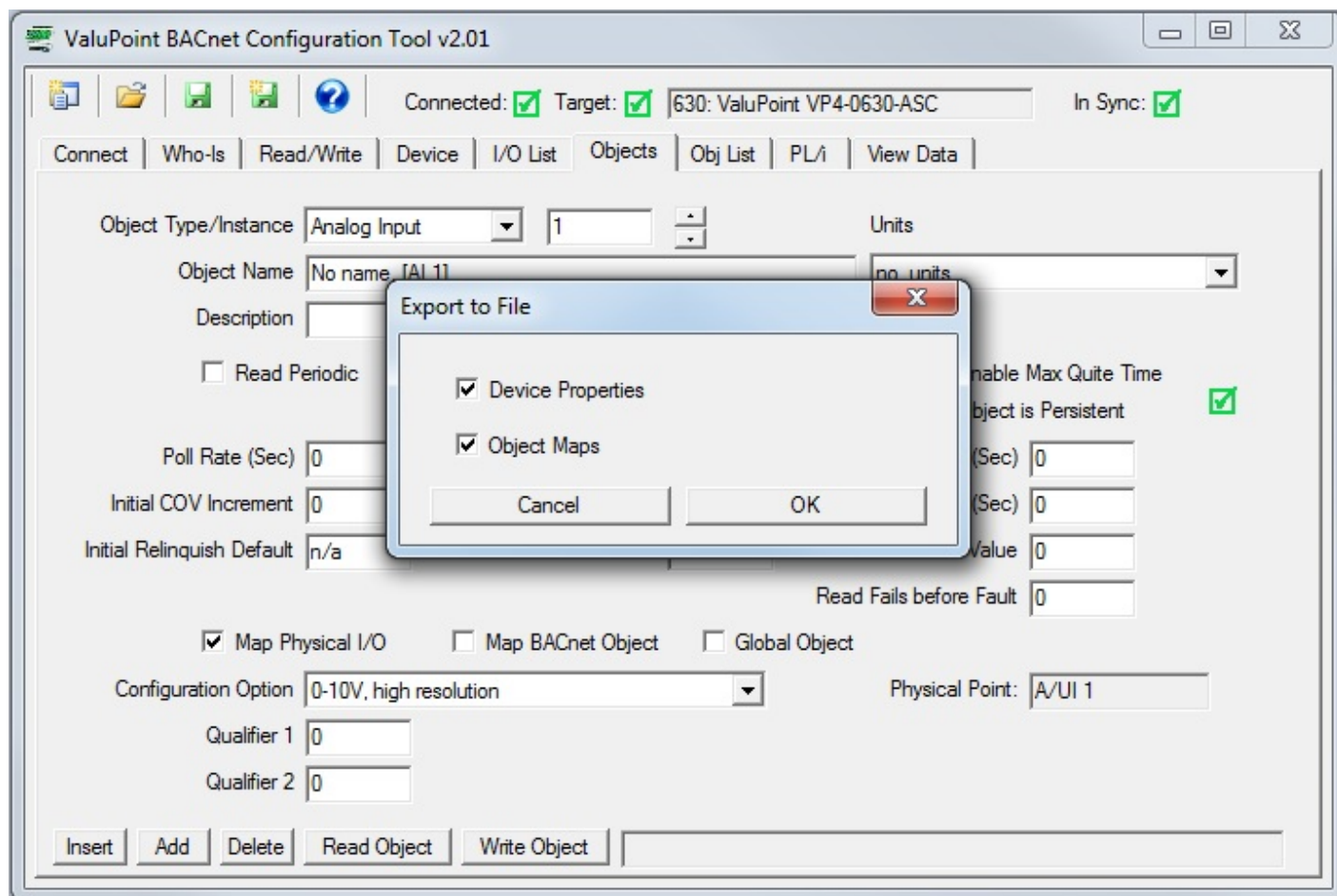
Click the 'new' icon to erase all object maps.

Click the 'open file' icon to read a set of object maps from an XML file. If you will be replicating the same configuration multiple times, you will save time by creating a file that can be reloaded. The currently open file will be indicated in the title bar of the program.

Click the 'save' icon to re-save the previously opened file, or to save for the first time if creating a new file. The presently displayed device configuration will be saved.

Click the 'save as' icon to create a copy of a previously opened file. A file dialog will appear allowing you to browse a selected directory and rename the file.

The file icons at the top of the tool screen will be different colors on different pages, and will be absent on some pages. These icons only appear where a file save is applicable. The different colors are meant to suggest different file types are used with different pages. There are 2 file types: (i) Device and Object Map XML, (ii) Object Map CSV. The XML file type will be saved from either the Device or Objects tabs while the CSV file type will be saved from the Obj List page.



When opening or saving an XML file, the dialog illustrated above will appear allowing you to select device properties, object maps, or both. If you wish to replicate only the object mapping in another device without overwriting the other device properties, then you would select only the object maps.

The original ValuPoint configuration tool saved two different XML files, one for device properties and one for object maps. They have now been combined into one file. You may read the older individual files into the new tool. Simply select device properties or object maps when reading the old files.



## 9 Object Map List

The screenshot shows the ValuPoint BACnet Configuration Tool v2.01 interface. The window title is "ValuPoint BACnet Configuration Tool v2.01". The status bar indicates "Connected: [checked] Target: [checked] 630: ValuPoint VP4-0630-ASC In Sync: [checked]". The main menu includes "Connect", "Who-Is", "Read/Write", "Device", "I/O List", "Objects", "Obj List", "PL/I", and "View Data". The "Read All" button is highlighted, and the "Write All" button is also visible. The "Objects" pane shows a list of objects with the following data:

Obj...	Name	R/W	Device	Object	Type	Configuration
AI 1	No name. [AI 1]	---	Local	AI [1]	A/UI #1	0-10V, high resolution
AI 2	No name. [AI 2]	---	Local	AI [2]	A/UI #2	0-10V, low resolution
AI 3	Remote AI 1	R	MS/TP [30...	AI [1]	Present_Value	
AI 4	Remote AI 2	R	MS/TP [30...	AI [2]	Present_Value	
AI 5		---	None	None	None	
AI 6		---	None	None	None	
AI 7		---	None	None	None	
AI 8		---	None	None	None	
AI 9		---	None	None	None	
AI 10		---	None	None	None	
BI 1	No name. [BI 1]	---	Local	BI [1]	BI/DI #1	
BI 2	No name. [BI 2]	---	Local	BI [2]	BI/DI #2	
BO 1	No name. [BO 1]	---	Local	BO [1]	BO/DO #1	
BO 2	No name. [BO 2]	---	Local	BO [2]	BO/DO #2	

### 9.0 Do This for First Time Startup

Click 'Read All' to read the entire object map list from the device. You must have already read the device information on the Device page, namely to get the allocated object counts. Without the object counts, the tool does not know how many, and what type, object maps to ask for from the device.

Note: This has already been done if you clicked the 'Read All Config' button on the Who-Is page.

### 9.1 Read All, Write All

Click 'Read All' to begin the process of reading all of the available object maps from

the device. If the tool was not in sync with the device previously, the red dots in the first column will turn green as the map on that line is updated and synced with the device.

Click 'Write All' to begin the process of writing all object maps to the device. It would generally be assumed that you will only click this button after opening a file containing a previously defined configuration.

## 9.2 Select for Edit

Click any line in the list of objects to be taken directly to the object map editing page for that object.

## 9.3 File Read/Write (Object Map CSV)

Click the 'new' icon to erase all object maps.

Click the 'open file' icon to read a set of object maps from a CSV file. If you will be replicating the same configuration multiple times, you will save time by creating a file that can be reloaded. The currently open file will be indicated in the title bar of the program.

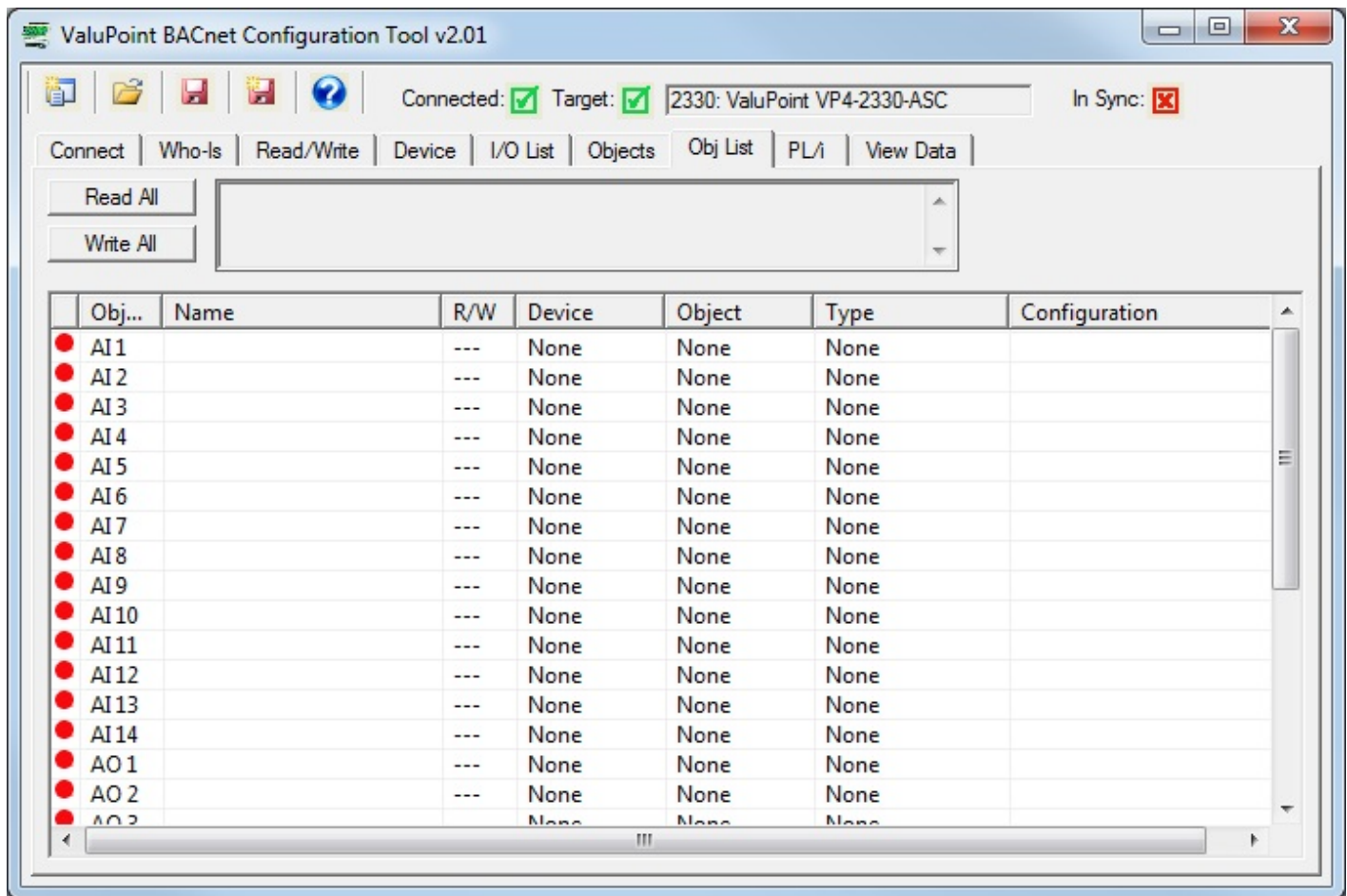
Click the 'save' icon to re-save the previously opened file, or to save for the first time if creating a new file. The presently displayed device configuration will be saved.

Click the 'save as' icon to create a copy of a previously opened file. A file dialog will appear allowing you to browse a selected directory and rename the file.

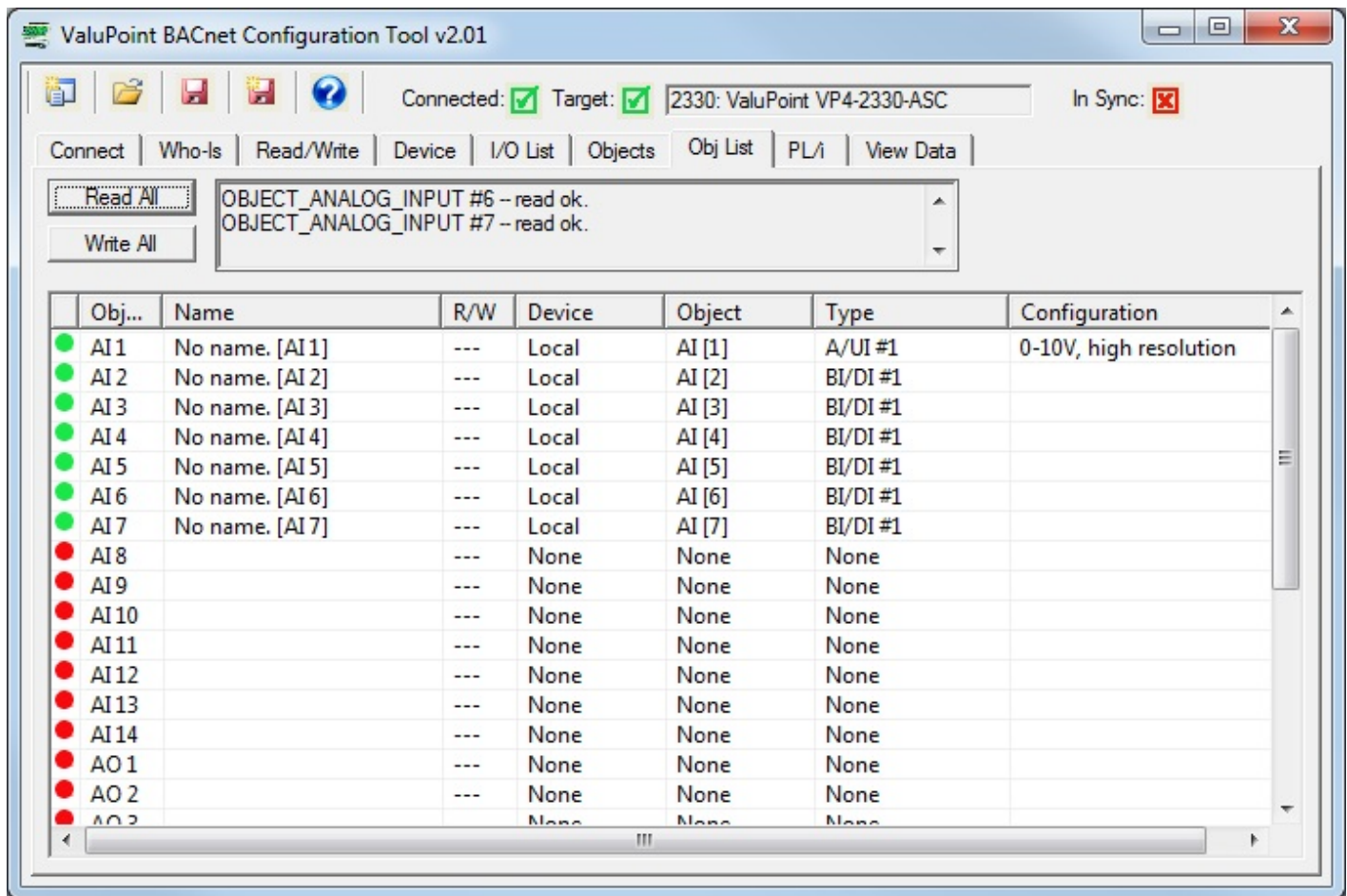
The file icons at the top of the tool screen will be different colors on different pages, and will be absent on some pages. These icons only appear where a file save is applicable. The different colors are meant to suggest different file types are used with different pages. There are 2 file types: (i) Device and Object Map XML, (ii) Object Map CSV. The XML file type will be saved from either the Device or Objects tabs while the CSV file type will be saved from the Obj List page.

NOTE: The CSV file must be in a specific format. The CSV file can be edited with any standard spread sheet program such as Microsoft Excel®. When creating or manipulating a large configuration, this can be a time saving approach to creating the configuration. However, some practice will be required to learn proper formatting, etc, before it will become a time saver. Refer to the section 'Editing a CSV Object Map File' that follows later.

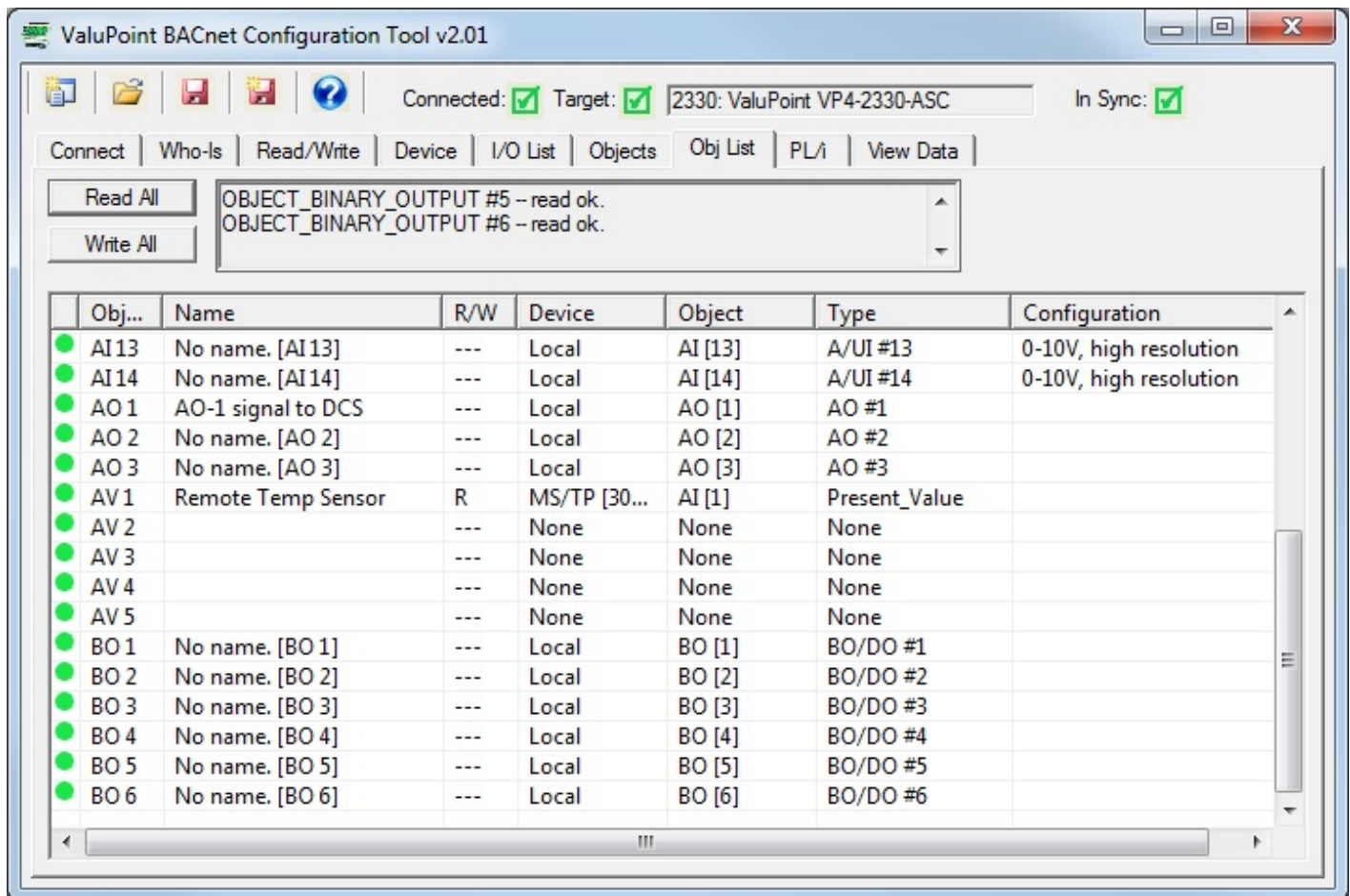
Prior to reading any objects, the screen will appear as follows:



During the process, the screen will appear as follows:



When complete, the screen will appear as follows. Physical I/O points on the ValuPoint will be listed as 'Local' in the Device column. When an object is mapped via the MS/TP Client to another BACnet device, the notation 'MS/TP' and the device instance will be indicated in the Device column.



If the configuration tool encounters an error while reading configuration from a ValuPoint, an error message will be displayed. The screen shot below shows an example. This particular error message should be a clue to check the model selected on the Connect page. If you are reading configuration from a VP4-0630 but have VP4-2330 selected on the Connect page, then you will be attempting to read configuration for an I/O point that does not exist once you get past point number six. If you see this error message, check the model on the Connect page, and re-read the configuration if model is changed.

ValuPoint BACnet Configuration Tool v2.01

Connected:  Target:  630: ValuPoint VP4-0630-ASC In Sync:


Connect | Who-Is | Read/Write | Device | I/O List | Objects | Obj List | PL/I | View Data

Read All Write All

OBJECT\_BINARY\_OUTPUT #1 - read ok.  
OBJECT\_BINARY\_OUTPUT #2 - read ok.

Obj...	Name	Configuration
AI1	No name. [AI 1]	0-10V, high resolution
AI2	No name. [AI 2]	0-10V, low resolution
AI3	Gateway AI 1	Value
AI4	Gateway AI 2	Value
AI5		
AI6		
AI7		
AI8		
AI9		
AI10		
BI1	No name. [BI 1]	---
BI2	No name. [BI 2] Unused	---
BO 1	No name. [BO 1]	Local BO [1] A/UI #5
BO 2	No name. [BO 2]	Local BO [2] A/UI #6

**Error!**

 Error: Invalid array index

OK





## 10 PL/i Programming for Control

 The screenshot shows the 'ValuPoint BACnet Configuration Tool v2.01' window. At the top, there are status indicators: 'Connected: [checked]', 'Target: [checked]', '630: ValuPoint VP4-0630-ASC', and 'In Sync: [checked]'. Below this is a menu bar with options: 'Connect', 'Who-Is', 'Read/Write', 'Device', 'I/O List', 'Objects', 'Obj List', 'PL/i', and 'View Data'. The main area contains several input fields and buttons:
 

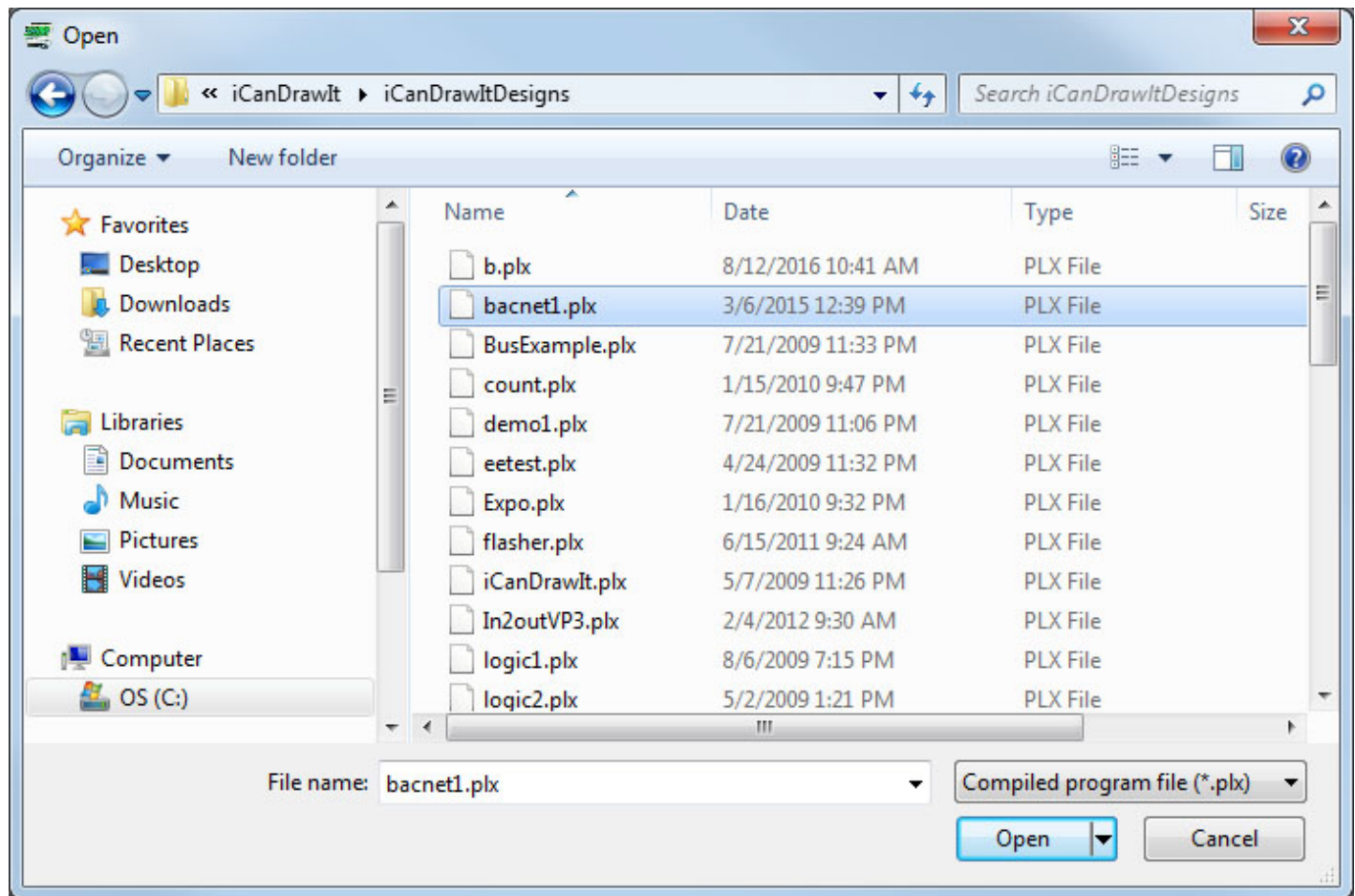
- 'Object Name' field with 'PL/i Program' entered and a 'Read' button.
- 'Program Change' field with '5'.
- 'Program State' field with '0'.
- 'Reason for Halt' field with '0'.
- 'Description of Halt' field with 'None'.
- 'Program Description' empty text area.
- 'Program File' empty text area.
- 'File Size' field with '0'.
- 'File Type' field with 'PLX Program Code'.
- 'Program Change' dropdown menu with 'Run Program' selected.
- 'Write' button.
- 'Send' button.

### 10.0 Skip This for First Time Startup

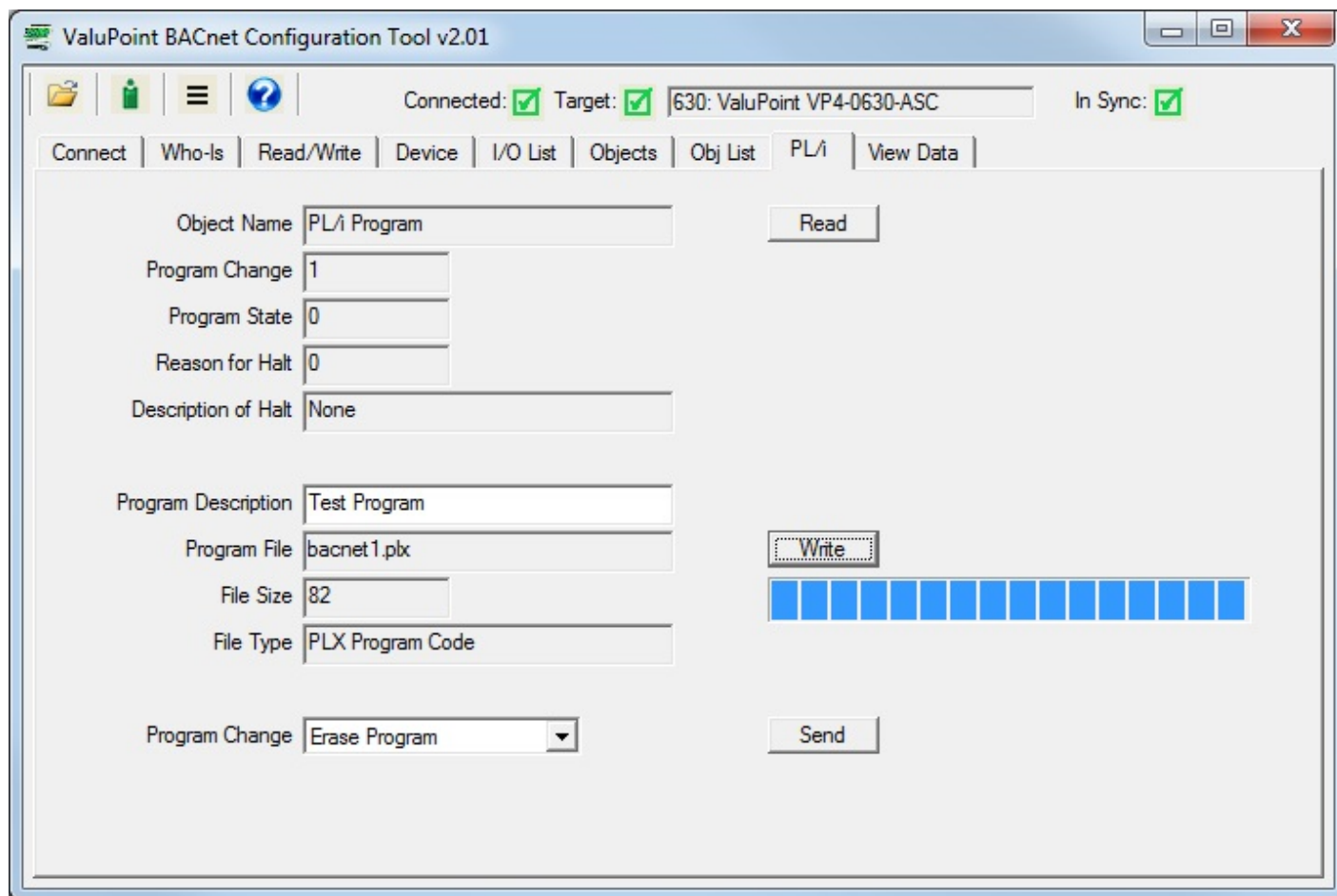
There is nothing that needs to be done on this page for initial startup. To use the ValuPoint as a simple slave I/O device, you will not need this page at all. If you want to turn the ValuPoint into a programmable controller, you will start here.

### 10.1 Program Loading and Execution

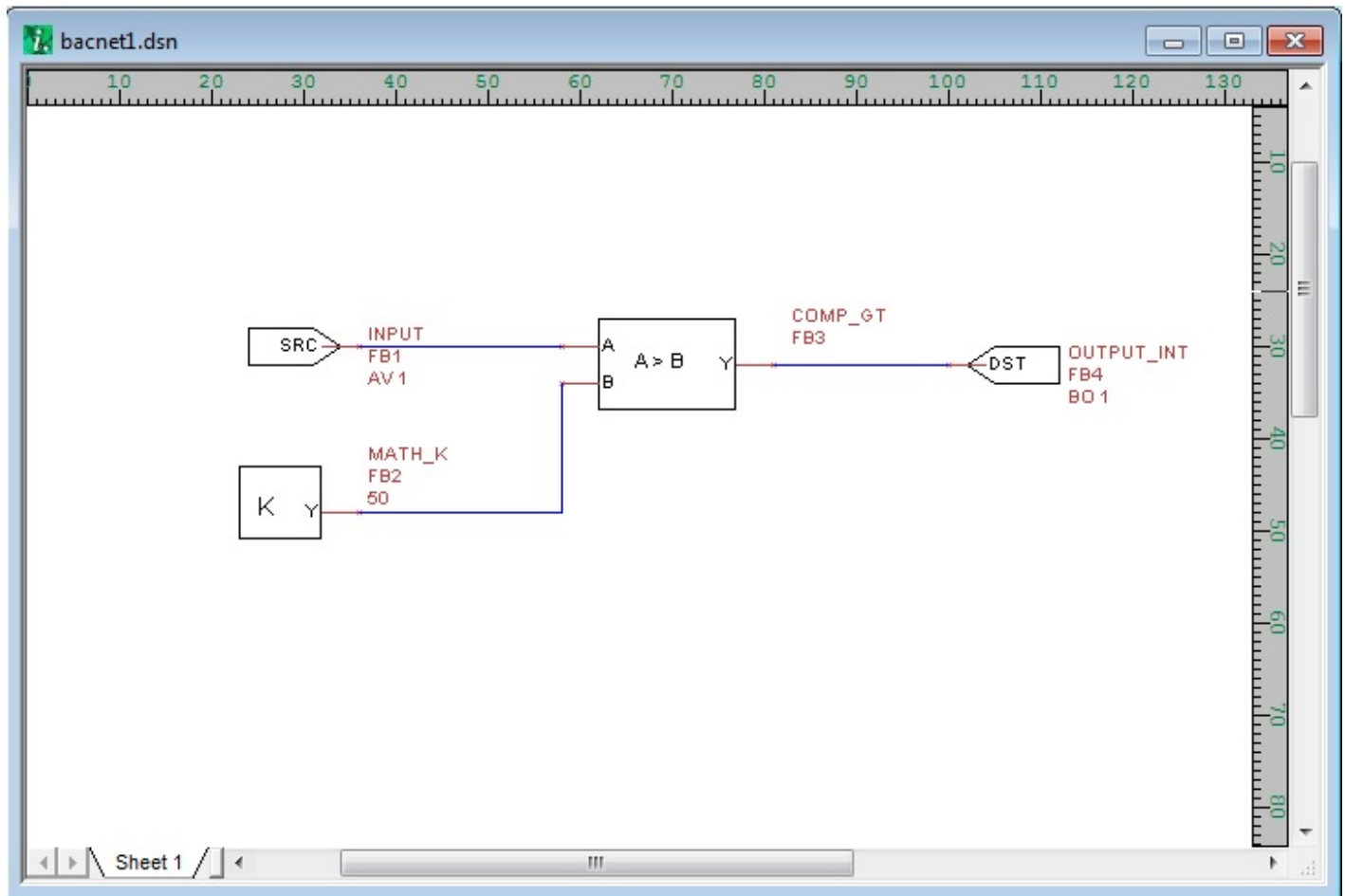
Click on the file folder icon at the top left to open a file. The file open dialog will appear. Select a .plx file from the list. If you do not yet have any programs compiled, you will need to use the program editing tools to create and compile a program.



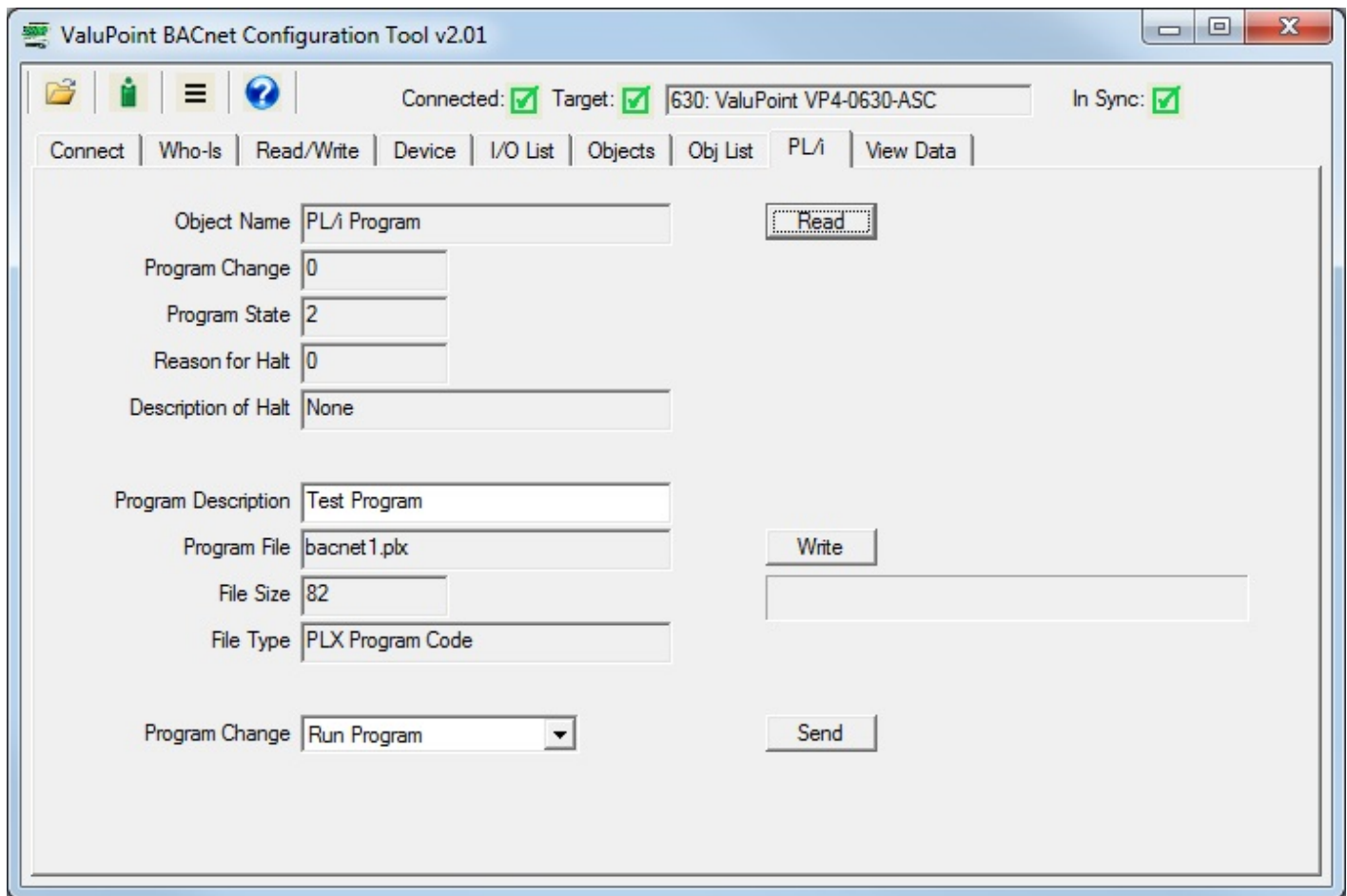
After a program (.plx file) has been opened, click the Write button to send that program to the ValuPoint. You can click the Read button at any time to see what program has been loaded, and what its execution state is. The screen shot below shows that 'bacnet1.plx' has been loaded, but it is idle.



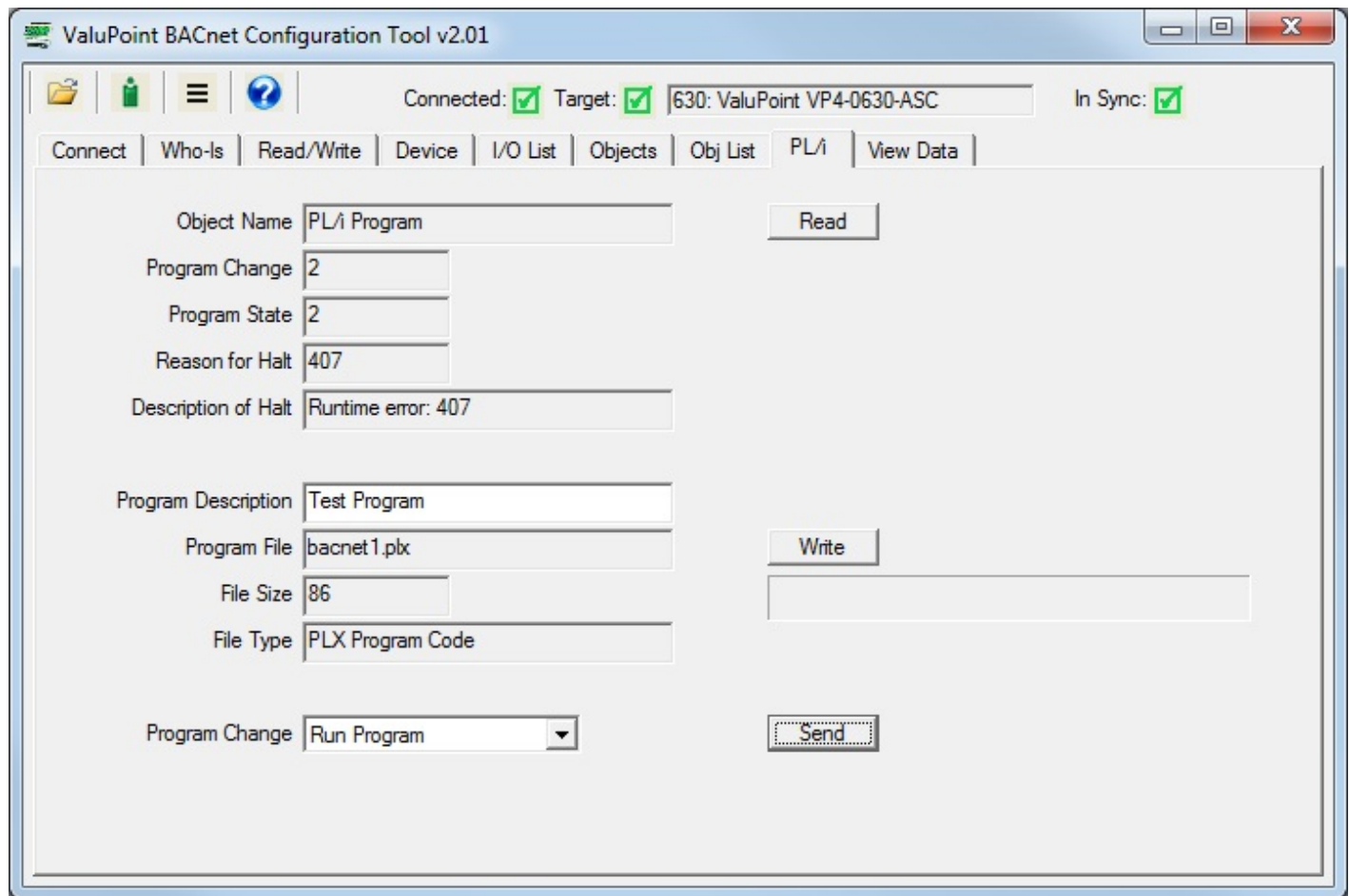
The example here shows that the program file size is 82 bytes. This example program is a very simple threshold program that will turn on the relay at BO 1 if the value of AV 1 exceeds a value of 50. The program as would be illustrated in i.CanDrawIt is shown below.



To invoke execution of the program, select 'Run Program' from the Program Change list, and then click Send.



If the program encounters a fatal error during execution, it will be reported as Reason for Halt as illustrated below. See section below for codes that may appear.



## 10.2 Program Editing and Debugging

Click the green "i" icon next to the folder icon to open the i.CanDrawIt graphical programming tool that was used to create the program illustrated above. It has its own set of help pages. Click on the "?" icon in that tool for more about programming.

Click the "line" icon next to the "?" icon to open the line programming tool. If you do not want to "draw" a program, but would rather write a program using the native PL/i programming language, you can do this. The line programming tool also has its own set of help pages.

The PL/i programming language is a derivative of PL/1 but is not the same as PL/1. The language is referred to as PL/i with "i" as in i.CanDrawIt.

## 10.3 Program Capacity

Maximum compiled program (.plx file) size: 65,280 bytes

Maximum RAM available for program variable and stack space: 8192 bytes

Maximum EEPROM available: 2048 bytes

## 10.4 Program States and Error Codes

BACnet Program Change codes:

- 0 = ready
- 1 = load
- 2 = run
- 3 = halt
- 4 = restart
- 5 = unload

#### BACnet Program State codes:

- 0 = idle
- 1 = loading
- 2 = running
- 3 = waiting
- 4 = halted
- 5 = unloading

#### Reason for Halt codes (BACnet Program Error):

- 0 = no error (not halted)
- 1 = program load failed
- 64 = normal stop, end of program reached
- 65 = external stop via Program Change
- 66 = debug execution, suspended
- +n = error code (400 or above)

#### Non-fatal runtime errors:

- 401: subscript out of bounds, non-fatal
- 402: divide by zero, non-fatal
- 406: EEPROM address out of range, operation skipped
- 407: object instance out of bounds, operation skipped

Note: Error codes will show up as "Reason for Halt" even if the error was not necessarily fatal. This is because "Reason for Halt" is the only available standard Program Object property whose purpose is to report errors. Check the Program State to determine if the program is actually halted.

#### Fatal runtime errors:

- 451: unrecognized opcode, fatal
- 452: stack overflow, fatal
- 453: stack underflow, fatal
- 454: program pc out of bounds, fatal



## 11 View Data

Obj...	R/W	Device	Object	Type/Prop	Present Value	Reliabil...	Status Flags
AI 9	---	Local	AI [9]	A/UI #9	0.006445	0	F,F,F,F{ }
AI 10	---	Local	AI [10]	A/UI #10	0.006445	0	F,F,F,F{ }
AI 11	---	Local	AI [11]	A/UI #11	0.006445	0	F,F,F,F{ }
AI 12	---	Local	AI [12]	A/UI #12	0.006445	0	F,F,F,F{ }
AI 13	---	Local	AI [13]	A/UI #13	0.006445	0	F,F,F,F{ }
AI 14	---	Local	AI [14]	A/UI #14	0.006445	0	F,F,F,F{ }
AO 1	---	Local	AO [1]	AO #1	0.000000	0	F,F,F,F{ }
AO 2	---	Local	AO [2]	AO #2	0.000000	0	F,F,F,F{ }
AO 3	---	Local	AO [3]	AO #3	0.000000	0	F,F,F,F{ }
AV 1	R	MS/TP [3010]	AI [1]	Present_Value	0.000000	0	F,F,F,F{ }
BO 1	---	Local	BO [1]	BO/DO #1	1	0	F,F,F,F{ }
BO 2	---	Local	BO [2]	BO/DO #2	0	0	F,F,F,F{ }
BO 3	---	Local	BO [3]	BO/DO #3	0	0	F,F,F,F{ }
BO 4	---	Local	BO [4]	BO/DO #4	0	0	F,F,F,F{ }
BO 5	---	Local	BO [5]	BO/DO #5	0	0	F,F,F,F{ }
BO 6	---	Local	BO [6]	BO/DO #6	0	0	F,F,F,F{ }

### 11.0 Skip This for First Time Startup

Nothing needs to be done on this page to get the tool initially synced to the device.

### 11.1 Read All

Click 'Read All' to cause the tool to query all BACnet objects in the ValuPoint device. A brief summary of the object's mapping is displayed along with its Present Value, Reliability code, and Status Flags. Progress of the 'Read All' process will scroll through the log window at the top.

### 11.2 Data Definitions



The Object column indicates the type and instance of BACnet object in the local ValuPoint.

The R/W column indicates the following: R means periodic read, W means periodic write, W+ means write on delta. RW+ means a combination of those, etc.

The Device column will show 'Local', or show what remote slave device this object is mapped to. 'Local' means a physical I/O point on the local ValuPoint. 'MS/TP[N]' means BACnet slave whose device instance is N.

The Object column is significant only when the object is mapped via the MS/TP client to a remote device. Object shows what object in the slave device is mapped to this local object. The object type and instance are shown, e.g. 'AI[1]' means Analog Input 1.

Type shows the physical I/O point locally, or the remote object property type, such as Present Value.

Present Value shows a representation of the present value of the BACnet object in the ValuPoint.

Reliability code for the BACnet object in the ValuPoint is shown. These are listed in a later section.

Status Flags shows the status flags for the object. The standard flags are: IN\_ALARM, FAULT, OVERRIDDEN, and OUT\_OF\_SERVICE respectively (i.e. last T/F is out of service flag). The flag is F to show false/0 and T to show true/1.



## 12 Object Properties

### 12.1 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 writable 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> no response (64) crc error (65) exception, illegal function code (66) exception, illegal data address (67) exception, illegal data value (68) exception, code+65, rarely used (69..79) configuration property fault (80) exception, code not recognized (81) BACnet client read/write timeout (82) BACnet client received error response from slave (83)
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)
<i>Vendor Specific Object Properties:</i>	
Object_Map_Usage (801) (W)	BIT STRING: (0) object is mapped to BACnet object on another device (1) object is mapped to local physical I/O (2) set default on power-up (3) set default on comm. fail (4) read periodic (5) write periodic (6) write on delta (7) enable max quiet (8) object value is persistent (9) object is reserved for global use

Poll_Period (802) (W)	Unsigned Modbus poll/update time in seconds
Slope (809) (W)	REAL BACnet = Modbus * slope + intercept Modbus = (BACnet – intercept) / slope
Intercept (810) (W)	REAL
Default_Value (811) (W)	REAL (analog objects) ENUMERATED (binary objects) Unsigned (multi-state objects)
Max_Quiet_Time (812) (W)	Unsigned
BACnet_Slave_Timeout (813) (W)	Unsigned
Max_Read_Fail_Count (814) (W)	Unsigned
Send_On_Delta (815) (W)	REAL (analog objects only)
Initial_COV_Increment (816) (W)	REAL (analog objects only)
Initial_Relinquish_Default (818) (W)	REAL (analog objects) ENUMERATED (binary objects) Unsigned (multi-state objects)
BACnet_Slave_Device (819) (W)	Unsigned
Slave_Object_Type (820) (W)	ENUMERATED
Slave_Object_Instance (821) (W)	Unsigned
Slave_Index (822) (W)	Unsigned (value is index + 1, 0=no index)
Slave_Priority (823) (W)	ENUMERATED (1..16)
Slave_Property_Type (824) (W)	ENUMERATED (see BACnet protocol specification for type codes)
Slave_Data-Encoding (825) (W)	ENUMERATED (0) Null, (1) Boolean, (2) Unsigned integer, (3) Signed Integer,

	(4) Real (floating point), (8) Bit string, (9) Enumerated
Slave _Bit_Position (826) (W)	ENUMERATED
Slave_Address (827) (W)	ENUMERATED (MAC address if BACnet slave does not support dynamic binding)
Error_Class (830)	Unsigned
Error_Code (831)	Unsigned
Config_Write_Command (832) (W)	ENUMERATED (1) write config to EEPROM
Hardware_Configuration (841) (W)	ENUMERATED
Hardware_Qualifier_1 (842) (W)	ENUMERATED
Hardware_Qualifier_2 (843) (W)	ENUMERATED
Feedback (848)	Unsigned
Raw_Data (849)	Unsigned

## 12.2 Device Object Properties

The following properties are found in the Device object of the ValuPoint. In addition to standard Device properties, configuration properties that apply at a system level to the whole device are included here.

Property	Encoding
Object_Identifier (75) (W)	BACnetObjectIdentifier
Object_Name (77) (W)	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) (W)	Unsigned
Number_Of_APDU_Retries (73) (W)	Unsigned
Device_Address_Binding (30)	List of BACnetAddressBinding
Database_Revision (155)	Unsigned
Max_Master (64) (W)	Unsigned
Max_Info_Frames (63) (W)	Unsigned
Location (58) (W)	CharacterString
Max_Segments_Accepted (167)	Unsigned
APDU_Segment_Timeout (10) (W)	Unsigned
Active_COV_Subscriptions (152)	List of BACnetCOVSubscription
Last_Restart_Reason (196)	BACnetRestartReason
Local_Time (57) (W)	Time
Local_Date (56) (W)	Date
UTC_Offset (119) (W)	INTEGER
Daylight_Savings_Status (24) (W)	BOOLEAN

Restart_Notification_Recipients (202)	List of BACnetRecipient
Time_Of_Device_Restart (203)	BACnetTimeStamp
<i>Vendor Specific Object Properties:</i>	
Physical I/O Assignment (881) (W)	BACnetObjectIdentifier Index: 1..MAX_HARDWARE_POINT
MSTP_Port_Baud_Rate (1201) (W)	ENUMERATED: (9600, 19200, 38400, 76800) NOTE: Changes to port settings will take effect only after a COLDSTART or WARMSTART command is issued.
MSTP_Station_ID (1202) (W)	Unsigned
Enable_Auto_Reset (1203) (W)	BOOLEAN (True if reliability codes should auto return to zero)
Reinit_Password (1205) (WO)	CharacterString Note: This property is write-only, and is only writable with unlock sequence.
Password_Unlock (1206) (W)	Unsigned Note: Must write device's own device instance to this property to unlock password for writing, and next property written must be 1205. Any other sequence will reset and lock out password write.
Device_Allocation (1251) (W)	Unsigned Index: Use object type analog-input (0) analog-output (1) analog-value (2) binary-input (3) binary-output (4) binary-value (5) multi-state-input (13) multi-state-output (14) multi-state-value (19) (value is count allocated of objects of type indexed) (attempting to write a count that is not acceptable will return error class Services, code Invalid Configuration Data)  NOTE: After writing all 9 object allocation counts,

a WARMSTART must be issued. All object maps will be cleared.



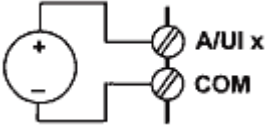
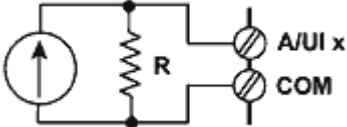
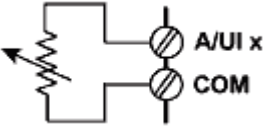


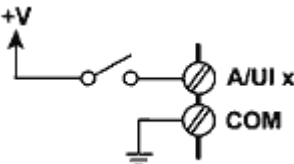
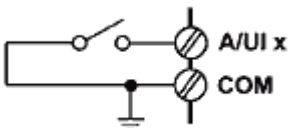
## 13 Physical I/O Connections

### 13.1 Connection of Inputs

The ValuPoint contains no configuration jumpers. There is no need to open the enclosure for configuration of I/O. Analog input types are switched between voltage and thermistor inputs under software control.

Input points should be connected as indicated in the various diagrams below. In addition to selecting a wiring diagram, the corresponding selections should be made on the I/O List page of the configuration tool.

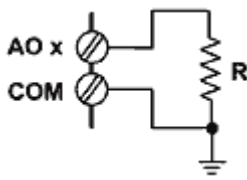
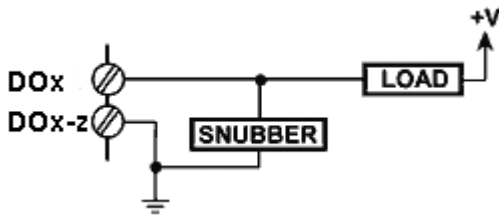
I/O Point Type	Wiring Guide	Additional Information
<b>A/UI Analog Input: 0-10VDC Voltage Input</b>		A/UI inputs 1-14 will accept voltage inputs of up to 10VDC. Voltages to 12VDC will be measured. Voltages to 24VDC will be tolerated, but measurement is internally limited to a reading of 12VDC.
<b>A/UI Analog Input: 0-20mA Current Input</b>		A/UI inputs 1-14 will accept current inputs of 0-20mA with the addition of a 500 ohm 1/2 watt external resistor. A 500 ohm resistor will produce a 0-10VDC signal. A 250 ohm resistor may also be used to produce a 0-5VDC signal.
<b>A/UI Analog Input: Thermistor Input</b>		A/UI inputs 1-14 will accept thermistors of 3k, 10k, or 20k ohms. Linearization via

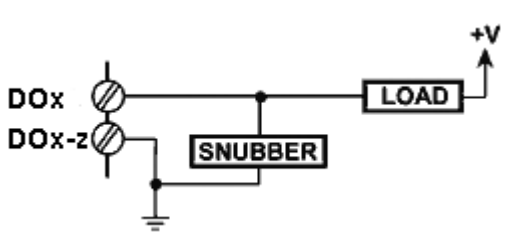
		interpolation of a 56-point table is performed internally.
<b>A/UI Discrete Input: Discrete Voltage</b>		A/UI inputs 1-14 may be connected as discrete voltage sensing inputs. Inputs up to 24VDC are tolerated, but threshold sensing only functions over the 0-10VDC range.
<b>A/UI Discrete Input: Dry Contact Closure to Ground</b>		A/UI inputs 1-14 may be connected as discrete inputs sensing dry contact closure to ground. Internal excitation of 5V @ 1mA is provided.

## 13.2 Connection of Outputs

The ValuPoint contains no configuration jumpers. There is no need to open the enclosure.

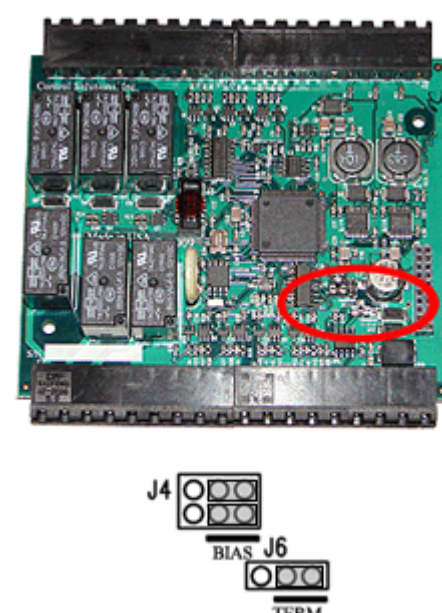
Output points should be connected as indicated in the various diagrams below. The discrete (relay) outputs are SPST N.O. with the common side banked in groups of 3. The DO1-3 and DO4-6 are not electrically common to each other or to COM/GND connections.

I/O Point Type	Wiring Guide	Additional Information
<b>Analog Output: 0-20mA Current Output</b>		AO outputs 1-3 produce outputs of 0-20mA. Use of an external 500 ohm resistor will produce a 0-10VDC voltage output. For best results, locate the resistor at the input of the receiving device.
<b>Discrete Output: Form A Relay AC Load</b>		BO/DO outputs 1-6 are Form A dry contact relays rated for 5A @ 120VAC (resistive). Snubbers should be used with inductive loads. Rating for inductive load is 2A @ 120VAC.

<p><b>Discrete Output: Form A Relay DC Load</b></p>		<p>BO/DO outputs 1-6 are Form A dry contact relays, also rated for 5A @ 30VDC (resistive). Snubbers should be used with inductive loads. Rating for inductive load is 2A @ 30VDC.</p>
---	---	---

### 13.3 Connection of Power and Communications

Power and communications should be connected as indicated below.

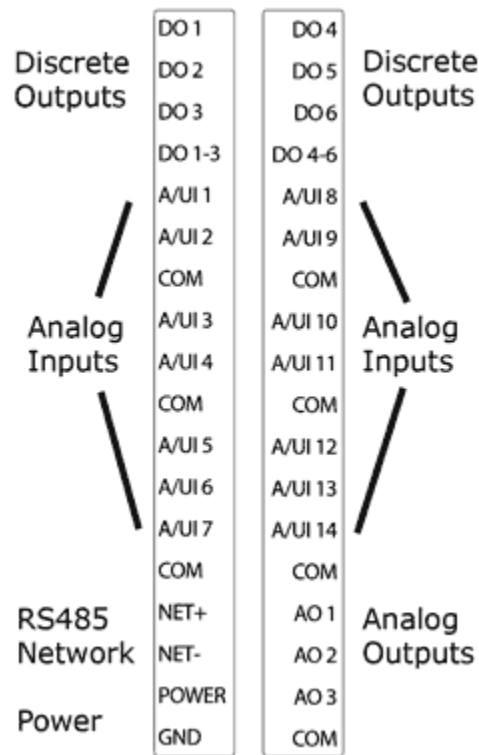
I/O Point Type	Wiring Guide	Additional Information
<p><b>Power</b></p>	<p>Connect AC or +DC power to Power terminal. Connect common or -DC power to GND terminal. GND is common to all terminals labeled COM.</p>	<p>Nominal power consumption is 6 watts, or 0.25A @ 24VDC, with all relays on and all analog outputs at 20mA.</p>
<p><b>Communications</b></p>	<p>Connect MS/TP RS-485 network to NET+/- terminals.</p>	<p>Communication signals comply with EIA-485 standard.</p>
<p><b>Line Conditioning</b></p>	<p>VP4-2330 Line Conditioning:</p>  <p>VP4-0630 Line Conditioning:</p>	<p>The ValuPoint is shipped with network line bias and termination disabled. If the ValuPoint will be the last device at the end of the RS-485 link, then termination should be enabled. Often it is also necessary to have line bias enabled, but this must only be done at one point on the network (not both ends like termination).</p> <p>The bias and termination jumpers are located in the area of the circuit board highlighted. The bias and termination are enabled when the shunts are aligned with the white bars next to the header pins. Move the shunts to the opposite end to disable. It is not necessary to enable</p>



both at the same time. (Illustration shows both enabled.)

To open the ValuPoint, use a very small screw driver to slide the cover up over the plastic tabs on the ends of the case. Do not push the tabs in as they are fragile. Slide the cover over the tabs by prying the cover slightly away from the tabs.

VP4-2330 Terminals:

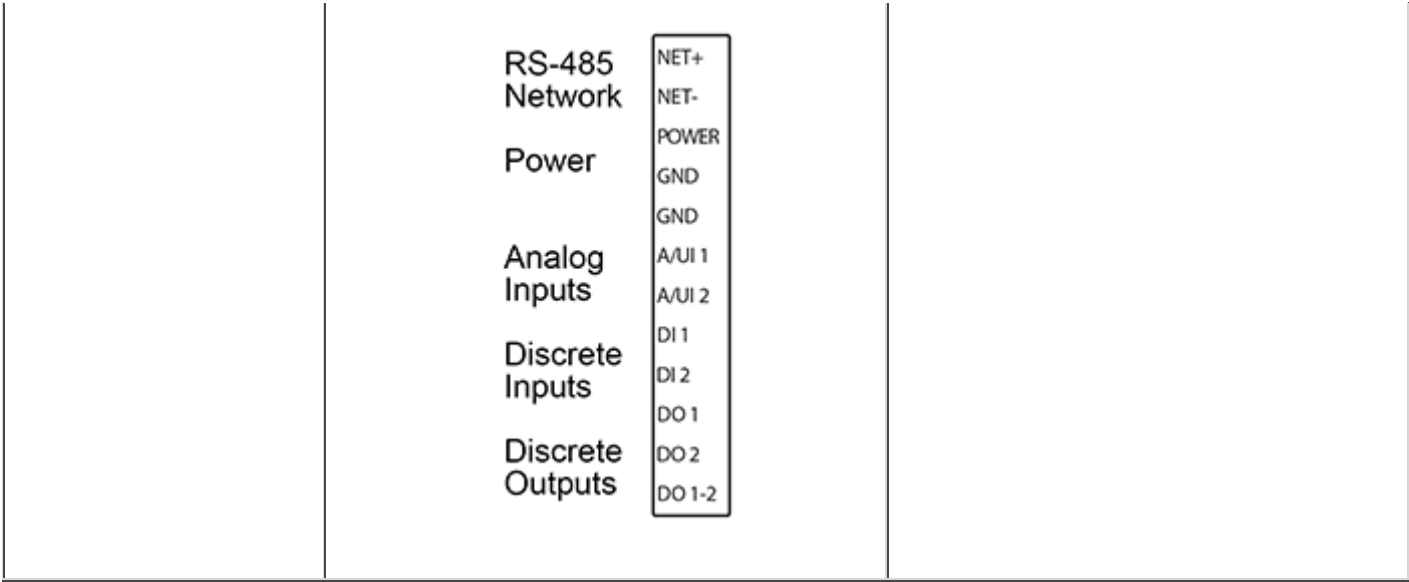


Screw terminals ratings are substantially in excess of any I/O point ratings.

Screw terminals are pluggable. They unplug from the unit in 9-position blocks. Any combination of smaller 5.00mm terminal blocks can optionally be used instead.

VP4-0630 Terminals:

**Wiring Terminals**





## 14 Trouble Shooting

### 14.1 Error Codes, Reliability Codes

Reliability codes (read via the Reliability property) are as follows:

- Configuration property fault (80)
- BACnet client read/write timeout (82)
- BACnet client received error response from slave (83)

If the object's reliability indicates code 83, meaning an error code was returned by a remote BACnet device, the actual code returned can be learned by reading the local object's properties 830 (class) and 831 (code).

### 14.2 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. When set, 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.

### 14.3 MS/TP Trouble Shooting - Connecting the USB Adapter

(a) Is the yellow LED next to the USB cable lit? If not, the PC does not recognize the adapter. Verify installation. Verify that MTX002 shows up as a Port in the Device Manager on your PC. The yellow LED next to the USB cable must be lit before continuing.

(b) Open the configuration tool. Select the port number indicated in the Device Manager. Select a known baud rate, max master setting, and unused Mac address. Click 'Connect'. Does the log window show "USB is responding"? If not, close the configuration tool, then disconnect and reconnect the USB cable to force a hard reset

of the adapter. Retry the 'Connect'. If you get an error message about the COM port, see that the supercom.dll is found in the same directory as the configuration tool, and recheck the port number on the PC's Device Manager. You must get "USB is responding" followed by "Ready" or "Auto connected" before continuing.

(c) Is the Token LED on the USB adapter flickering? If not, either there is nothing on the MS/TP network or the network has not recognized the adapter's reply to poll for master. If the network was online before the adapter was connected, the adapter will wait for the token indefinitely. If no other device on the network grants the token, the adapter will never begin communicating. Failure to get in sync, assuming all devices appear otherwise functional, is most often caused by either a mismatch in baud rate, mismatch in max master setting, or unusable choice of mac address.

(d) Is the PFM (poll for master) LED on the USB adapter nearly solid on? This means the adapter is busy sending out polls for master but not seeing any response. This can be caused by nothing being on the network, a mismatch in baud rate, or other low level communication problems. The normal LED activity when the adapter is connected and in sync with the network will be continuous flickering of the Token LED, usually but not always some periodic flickering of the PFM LED, and occasional flashing of the Data LED. The error LED could also flicker once in a while, but hopefully not often.

NOTE: The two "not really connected" indications you are most likely to see are (i) None of the LEDs next to the terminal block on the adapter ever flash; (ii) The PFM (yellow) LED is always flickering (nearly constant on). It is a somewhat random coincidence whether other devices on the network or the adapter gets ahead in polling. Devices on the MS/TP network always wait for the right opportunity to respond. If they never get that opportunity due to being out of sync as a result of incorrect baud rate, etc, they will appear to be "offline". Meanwhile, a device polling to look for other devices will constantly just poll because it is not getting any response. Because the device doing the constant polling will generate enough traffic on the line to make the line look busy to other devices, the other devices will never come online.

#### **14.4 MS/TP Trouble Shooting - Device Behavior**

The above discussion is focused on getting the USB to MS/TP adapter connected. However, the situation involving one device being out of sync with the network can happen with any device on the network. This is the situation that creates the appearance of a single device refusing to go online, or a single device taking down the network. If a device appears to work normally when connected to only the USB adapter, but either refuses to go online or appears to take down the network when connected to a running network, its port settings are out of sync with the rest of the network. It will take a few seconds for the network to re-sync any time a new device is added, but if the network does not return to normal token passing within a reasonable time, the newly added device needs attention.

Trouble shooting the ValuPoint MS/TP Connection

(a) Does the device have power? The blue LEDs inside the case, visible through the air vent slots, will both be on if there is power present.

(b) Are any LEDs flickering inside ValuPoint? The yellow poll-for-master LED and green token-pass LED inside the ValuPoint should be flickering some color rather constantly. Green indicates token passing, yellow indicates poll for master. Constant yellow (never green) means no other devices are responding. Constant off (both colors) means other devices are generating traffic on the network, but it is not being recognized by the ValuPoint as anything it can respond to. No token passing activity means either there are no other devices on the network, or there is a low level communications problem, namely one or more communication parameters are mismatched. There could also be a wiring problem. Check for reversed polarity on the data lines. If uncertain, just try swapping them – no physical harm will be done.

## 14.5 System Fault Indications

The red LED visible inside the ValuPoint case, viewed through the vent slots, is the system fault indicator. It will be on during initial power-up boot mode operation, but should otherwise be off except for indicating communications errors such as request for non-existent object.

Following a device reinitialize command from BACnet, the system fault LED will remain on for approximately 10 seconds, then turn off (or indicate system fault).

During normal operation, a watchdog timer is always running to force a soft restart the system in the event of a software hang. Should the system restart as a result of watchdog timeout, the system fault LED will flicker at a very rapid rate for approximately 10 seconds. An intentional software hang followed by watchdog timeout is used to force a restart upon receiving a restart command from BACnet. Therefore the indications are the same. However, if the restart indication is observed without sending the restart command from BACnet, please notify technical support.

System fault indication past the initial few seconds of boot mode followed by possible soft restart indication would consist of the red LED being mostly on, flashing off briefly some number of times, followed by a longer pause remaining on. Count the number of 'off' flashes. This is the fault code.

Fault codes are as follows:

- (1-6) Processor abort codes
- (7) System configuration CRC failed
- (8) Bad configuration parameter found in system configuration
- (9) EEPROM read failed
- (10) EEPROM write failed
- (11) EEPROM lock failed
- (12) Object allocation failed

Report any of these to technical support. There are both a primary copy and backup



copy of system configuration information. Both need to fail before the fault code will be indicated. These will generally indicate a hardware failure requiring factory attention. You should really never see any fault codes.

A processor abort will initially be indicated by the red LED on solid, and the yellow LED flashing a code from 1 to 6. However, the watchdog timer will normally restart the system sooner than you can observe this code and normal system fault indication will continue from that point. The abort cause is saved through soft restart. The term 'soft restart' means processor reset and complete reboot of the system as if power cycled. A power cycle is required for hard reset, and results in the same startup sequence except that processor abort codes are not retained.



## 15 Editing a CSV Object Map File

The easiest way to begin the process of configuring a device by editing a CSV file is to first create one using the configuration tool. Set up the device first. In particular, allocate the appropriate number of each object type on the Device page. Then go to the Object Map List page. Even though there will be an essentially blank list here, click the Save button to create a file with place holders for all of the objects. Then use a spread sheet program to modify the various entries to your liking.

Column A: Object – ASCII string identifying the local object type and instance, TT-N where N is instance 1..MAX\_OBJECT, and TT is one of the following:

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

Column B: IsPhysical – T if mapped to physical I/O point, or F

Column C: IsBACnet – T if mapped to BACnet, or F

Column D: IsGlobal - T if reserved for use without mapping, or F

Column E: IsPersistent - T if present value is preserved through power outage, or F

Column F: ReadPoll – T if periodic Read, or F

Column G: WritePoll – T if periodic Write, or F

Column H: WriteDelta – T if Write on Delta, or F

Column I: DefPOR – T if default on POR enabled, or F

Column J: DefNOK – T if default on comm. fail enabled, or F

Column K: MaxQuiet – T if max. quiet enabled, or F

Column L: PollTime – Integer, Periodic poll time in seconds

Column M: Timeout – Integer, BACnet slave timeout in seconds

Column N: MaxQuiet – Integer, max quiet time in seconds

Column O: Scale – Real, scale factor

Column P: Offset – Real, offset for scaling

Column Q: DefaultValue – Real or Integer as applicable to object type, default value

Column R: SendDelta – Real, delta threshold for Write on Delta

Column S: InitCOVincrement – Real, initial COV increment

Column T: InitRelinquish – Real or Integer as applicable to object type, initial relinquish default

Column U: FailCount – Integer, count of comm. fails before Fault indicated

Column V: Units – Integer, Units code as defined by BACnet standard

Column W: ObjectName – Character String

Column X: HwCfg - Hardware configuration code if object is mapped to physical I/O (see Section 8 for codes)

Column Y: HwQual1 - Hardware configuration qualifier value #1

Column Z: HwQual2 - Hardware configuration qualifier value #2

Column AA: HwPoint - Physical point number mapped to this object

Column AB: ObjDevice – Integer, BACnet device instance to read/write when mapped via BACnet Client

Column AC: ObjType – Integer, BACnet object type code as defined by BACnet standard

Column AD: ObjInstance – Integer, BACnet object instance to read/write via client

Column AE: ObjProperty – Integer, BACnet property code as defined by BACnet standard

Column AF: ObjIndex – Integer, array index, -1 if no index

Column AG: ObjPriority – Integer, priority 1-16 or 0 if none

Column AH: ObjEncoding – Integer, object encoding  
(0) Null

- (1) Boolean
- (2) Unsigned Integer
- (3) Signed Integer
- (4) Real
- (8) Bit String
- (9) Enumerated

Column AI: BitNumber – Integer, bit number



## Appendix A USB Driver Installation

The required USB driver used to be included as a standard part of Windows, and all that needed to be done to “install” the driver was provide a configuration file telling Windows which driver to use. Starting with Windows 7, that driver was no longer included, and Windows 8 complicated matters even further.

Control Solutions has licensed a USB driver and installer from a software development company that specializes in USB drivers. Drivers are all they do and they do it well, so we feel confident in our choice. Control Solutions paid a significant license fee so that we are able to provide it to our customers at no charge.

The USB driver provided in Control Solutions’ driver package will install in Windows XP and up, and both 32-bit and 64-bit versions. It includes the necessary driver signing verified through Verisign (now part of Symantec). This driver has been tested on Windows XP through Windows 10.

The driver package will show up as a zip file named “csimnUSB.zip”.



Unzip the contents of this file into a directory somewhere on your PC. The contents will look like this:

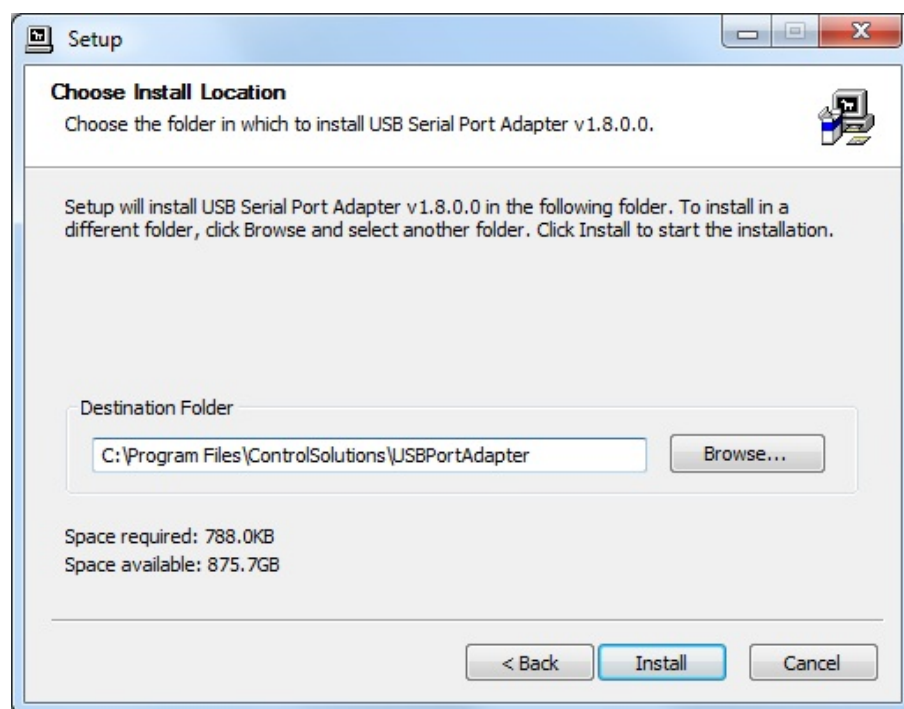


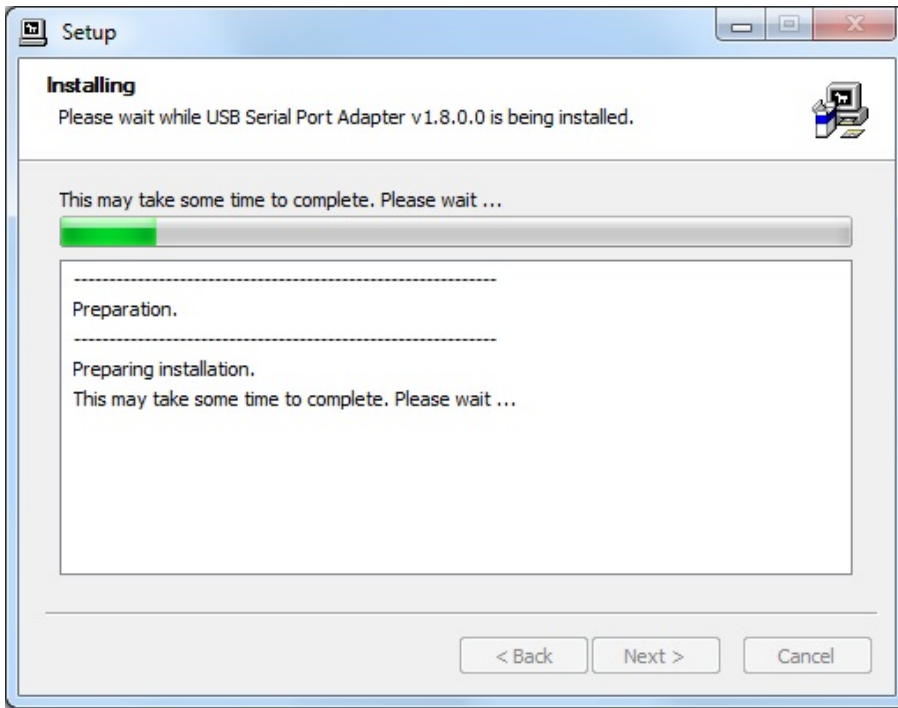
Double click “setup.exe”. Say “yes” to any questions about whether to trust this software. Also, for Windows 8, you should right click on the setup file and “Run as administrator” - you will need to be logged in with administrator privileges.

A sample of the series of screens you will see appears below. Basically all you need to do is follow the prompts, and click “yes”, “next”, “continue anyway”, etc, as applicable.

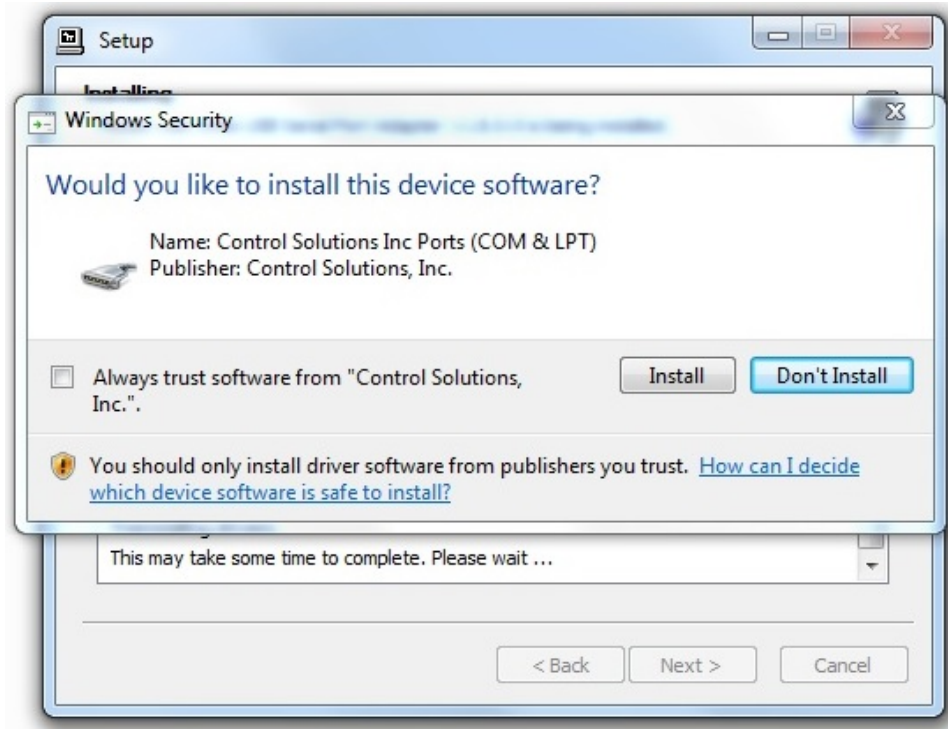
Technically, what you are doing in the process illustrated in the screen shots below is “driver

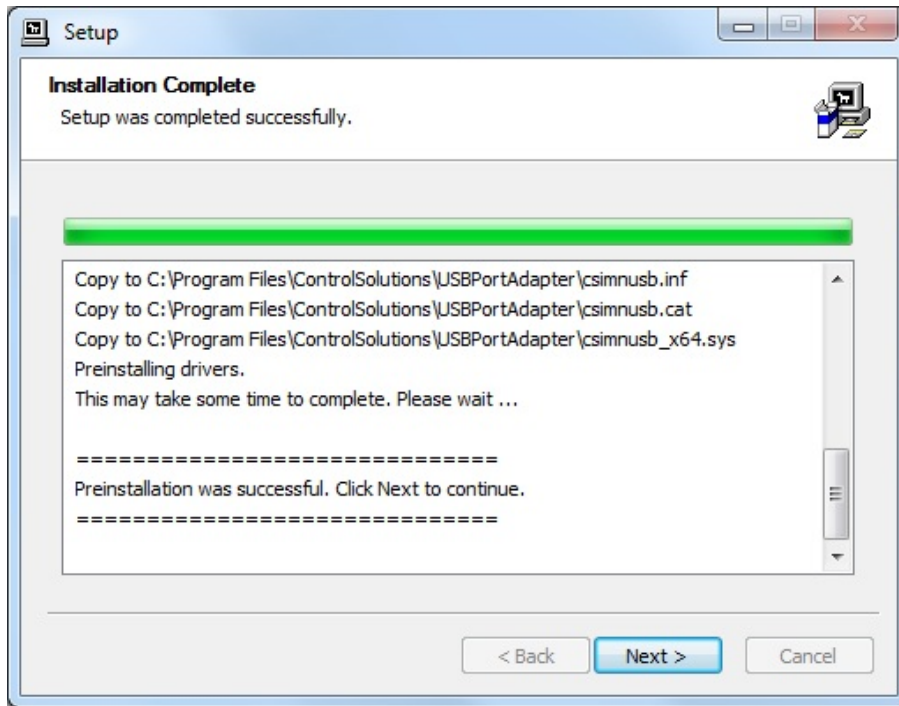
pre-installation". After initial installation of this package, the device will automatically find the right driver when you plug it in, and driver installation will be finalized. Windows 8 will install the driver quietly and usually say nothing about it. Windows 7 will display a prompt telling you the new device was installed, but will not require responding to any prompts. Windows XP will go through the characteristic "Found new hardware" routine with a series of dialogs and prompts the first time you plug the device in. Tell your PC "no" to searching the Internet, but "yes" to installing automatically, and "continue anyway" when it complains about Windows logo certification. (Windows 7 and 8 will not register any such Windows logo complaint.)





Click on "Install" when you get to this window:





When you get to this screen, you're done. Now plug in your USB device (MTX002, iReport, BB2-LON), allow the PC to finalize installation, and then go to the Device Manager via your PC's control panel to see which port the USB device got assigned to. Select this "COM" port in the Control Solutions configuration tool's "Connect" page.







## Appendix B – BACnet Property Type Codes

BACnet property type codes may be found in your copy of the BACnet protocol specification, ANSI/ASHRAE Standard 135-2008. 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;
```