



## User Guide

**Model VP6-1430  
ValuPoint VI  
Programmable I/O  
for BACnet MS/TP**

**Rev. 1.0 – Jan. 2024**

### VP6-1430 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

#### [3 Local Port Page](#)

- 3.1 Connect with BACnet MS/TP
- 3.2 Connect with BACnet IP
- 3.3 Auto Connect (MS/TP)

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

- 4.1 Finding Devices
- 4.2 Select Target
- 4.3 Getting Device Information
- 4.4 Clear Who-Is Cache
- 4.5 Read All Configuration
- 4.6 Who-Is via IP Router
- 4.7 Targeted Who-Is

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

- 5.1 Read Property
- 5.2 Write Property

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

- 6.1 Retrieve Device Object Information from Device
- 6.2 Change Device Object Configuration
- 6.3 Reassign Device Instance
- 6.4 Get Object Counts from Device
- 6.5 Reconfigure Object Counts
- 6.6 Fixing "Invalid Configuration"
- 6.7 Export and Import Configuration with XML File

#### [7 Object Map Page](#)

- 7.1 Common Parameters
- 7.2 Physical Hardware I/O Parameters
- 7.3 BACnet Client Parameters
- 7.4 Global Object
- 7.5 Read/Write Device
- 7.6 Export and Import Configuration with XML File

#### [8 Object Text Page](#)

- 8.1 Read/Write Object Text
- 8.2 Edit Object Text
- 8.3 Export and Import Text as CSV file
- 8.4 Clearing Object Text

#### [9 Object List Page](#)

- 9.1 Read/Write All Object Maps in Device
- 9.2 Select an Object Map for Editing
- 9.3 Export and Import Configuration as CSV file
- 9.4 Clearing Object Maps, Counts, and Device

## [10 Data List Page](#)

- 10.1 Read All Object Data
- 10.2 Error Codes

## [11 Programming Page](#)

- 11.1 Program Loading and Execution
- 11.2 Program Editing and Debugging
- 11.3 Program Capacity
- 11.4 Program States and Error Codes

## [12 Firmware Update Page](#)

- 12.1 Firmware Organization
- 12.2 Verifying Primary Application Image
- 12.3 Firmware Update Process

## [Appendix A Hardware Details](#)

- A.1 Connection of Inputs
- A.2 Connection of Outputs
- A.3 Connection of Power and Communications
- A.4 RS-485 Line Termination
- A.5 LED Indicators
- A.6 Restore Factory Defaults

## [Appendix B BACnet Object Properties](#)

- B.1 Data Object Properties (Analog, Binary, Multi-state)
- B.2 Device Object Properties

## [Appendix C CSV File Format](#)

- C.1 Object Maps
- C.2 Object Text

## [Appendix D Trouble Shooting](#)

- D.1 Observing BACnet Errors, LED Indicators
- D.2 Error Codes, Reliability Codes
- D.3 Auto-Reset Errors
- D.4 Trouble Shooting Tips, USB Adapter
- D.5 Trouble Shooting Tips, ValuPoint Behavior
- D.6 System Fault Indications

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

- E.1 BACnet Object Property Codes
- E.2 BACnet Engineering Units Codes
- E.3 BACnet Error Codes

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

- F.1 Driver Installation



# User Guide

## ValuPoint® VI



### Model VP6-1430 Programmable I/O for BACnet MS/TP Networks Rev. 1.0 – Feb. 2024

© 2024 Control Solutions, Inc.

## 1 Overview

### 1.1 How to Use This Guide

Section 1 gives an overview of the VP6-1430 programmable I/O device. Section 2 talks about installing the configuration software and connecting the VP6-1430. Sections 3 through 12 are guides for each of the tabs found on the screen of the configuration software. Appendix A through F are reference material.

### 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.**

**CAUTION: The lithium battery contained in this device may explode if mistreated. DO NOT recharge, disassemble, or dispose of in fire.**

**No action is required of the user to activate the battery that backs up the real time clock. Important: Replace battery with BR1225A only. Use of another battery may present a risk of fire or explosion.**

### **1.3 Overview of the VP6-1430**

Control Solutions' Model VP6-1430 ValuPoint® Programmable I/O for BACnet MS/TP operates as an auxiliary device for I/O expansion, or as a controller with soft PLC capability. This 5th generation ValuPoint® platform features 12 universal context sensitive inputs. When configured as an analog input, the A/D converter produces 12-bit resolution. When configured as a discrete or pulse input, inputs are treated as state inputs. The VP6-1430 also includes 2 Form A relay outputs. High speed pulse counting capability is provided on 4 inputs.

#### **Hardware Features of Model VP6-1430**

- 12 Analog/universal inputs, software selectable types
  - 0-10VDC, thermistor, discrete, dry contact, pulse
  - 0.1% reference, 12-bit resolution
  - Non-volatile totalizing count inputs  
(to 2Hz on all channels, to 1kHz on 4 channels)
- 2 Discrete outputs
  - Form A relay
  - 2A @ 120VAC
  - 2A @ 30VDC
- Battery backed real time clock/calendar
- BACnet MS/TP at 9600, 19200, 38400, 76800 baud
- Isolated RS-485 port
- 512KB non-volatile EEPROM configuration file capacity
- 64K Flash for User Program
- ARM 32-bit processor, 1MB Flash, 256K RAM
- Powered by 18-30VDC or 24VAC 50/60 Hz Class 2, 0.3A max.
- DIN rail mounting, 100mm H x 105mm W x 60mm D
- -40C to +80C, 5%-95% RH non-condensing
- Certifications: FCC, CE, UL 916 Listed

The VP6-1430 is configured by writing various special BACnet object properties. The VP6-1430 configuration tool simplifies this process by providing a graphical tool for automatically writing all of the necessary properties by selecting various options on the screen and then clicking the 'Write' button. Configuration is saved in non-volatile memory. 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.

## 1.4 Warranty

**This configuration 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.

**Warranty:** All Control Solutions products are warranted against defects in materials and workmanship for a period of time from date of shipment from factory as follows: Two years on non-mechanical parts, one year on mechanical parts (e.g. relays). Defective units will be repaired or replaced, at manufacturer's discretion, at no cost to user except when negligence or improper use has resulted in damage. The express warranty stated herein is in lieu of all other warranties, express or implied, including without limitation any warranties of merchantability or fitness for a particular purpose and all other warranties are hereby disclaimed and excluded by Control Solutions, Inc.

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

## 1.5 Required License Information

The VP6-1430 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

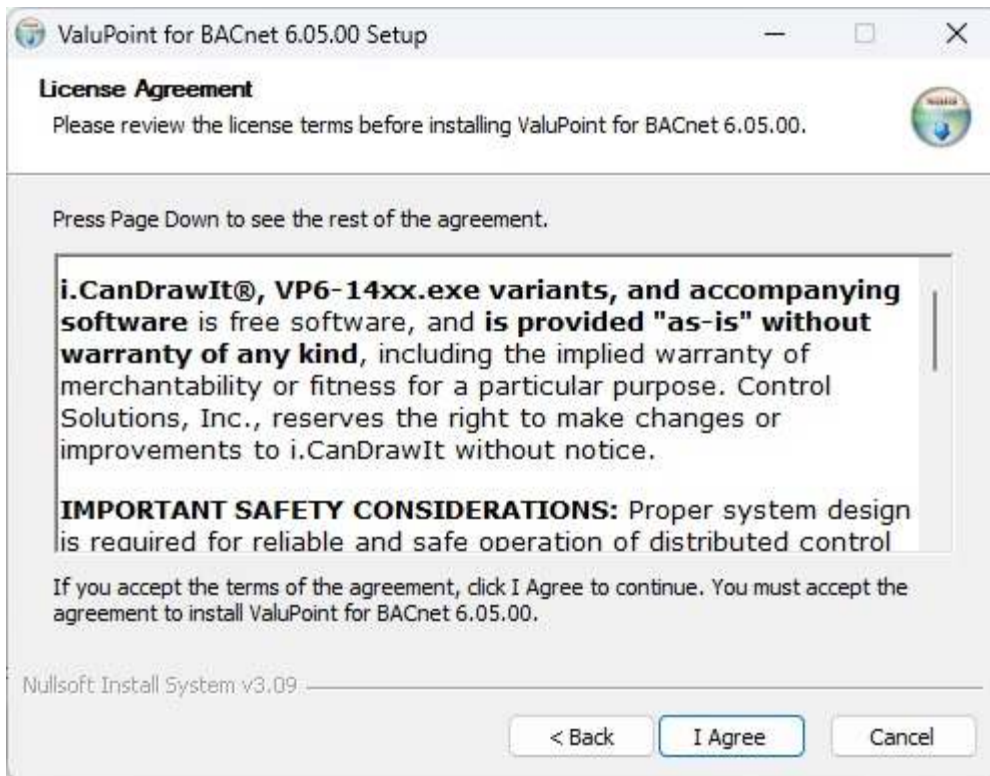
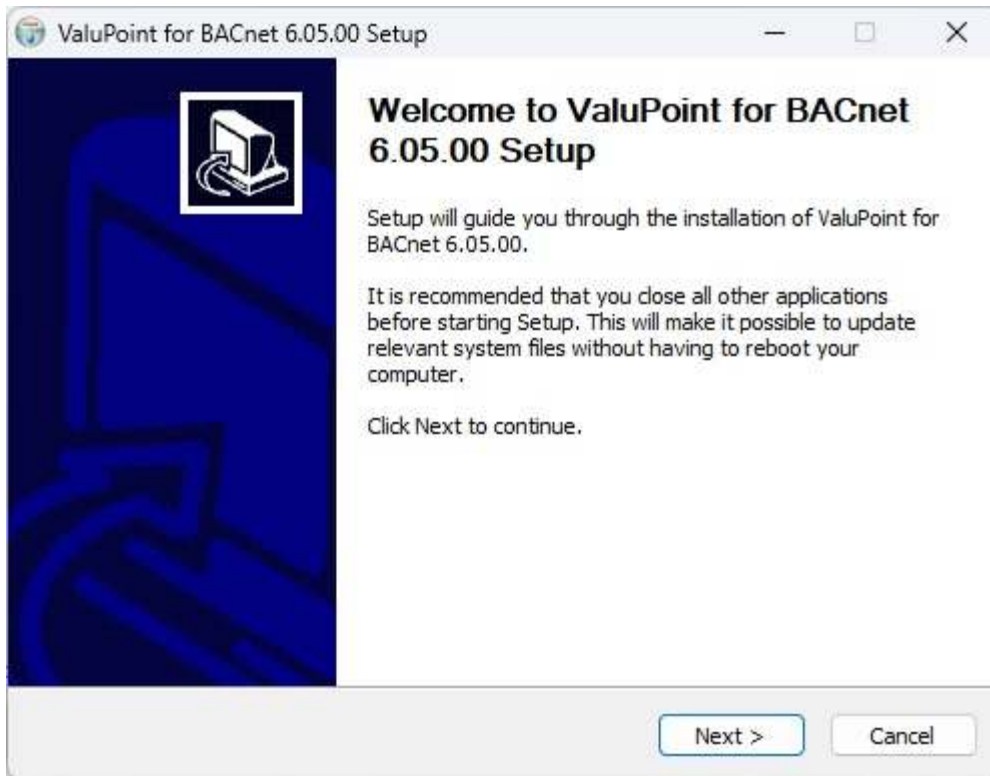
### 2.1 Installing the Configuration Software

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:

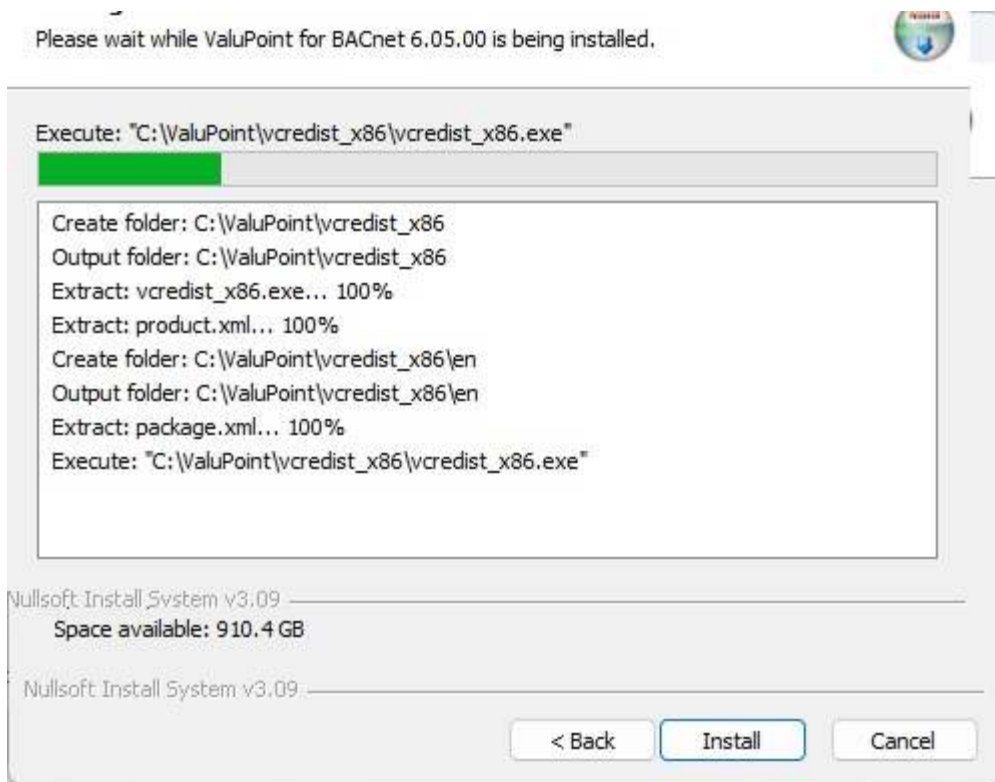
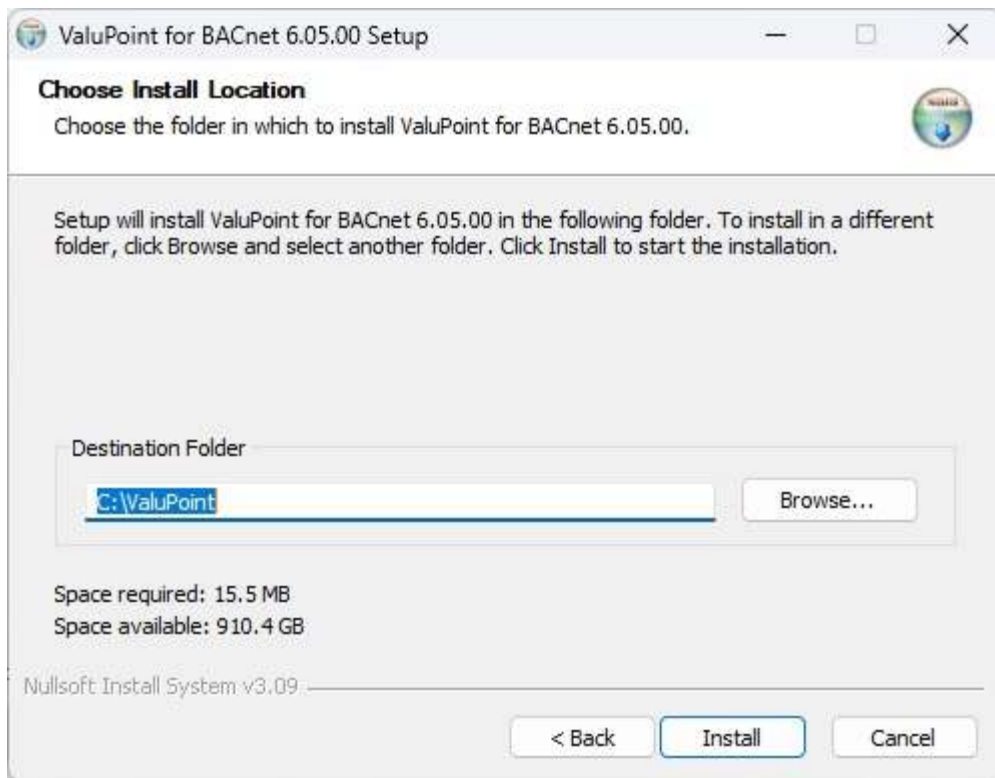


The installation is a 2-step installation. Install VP6\_1430 first. Then install i.CanDrawIt second.

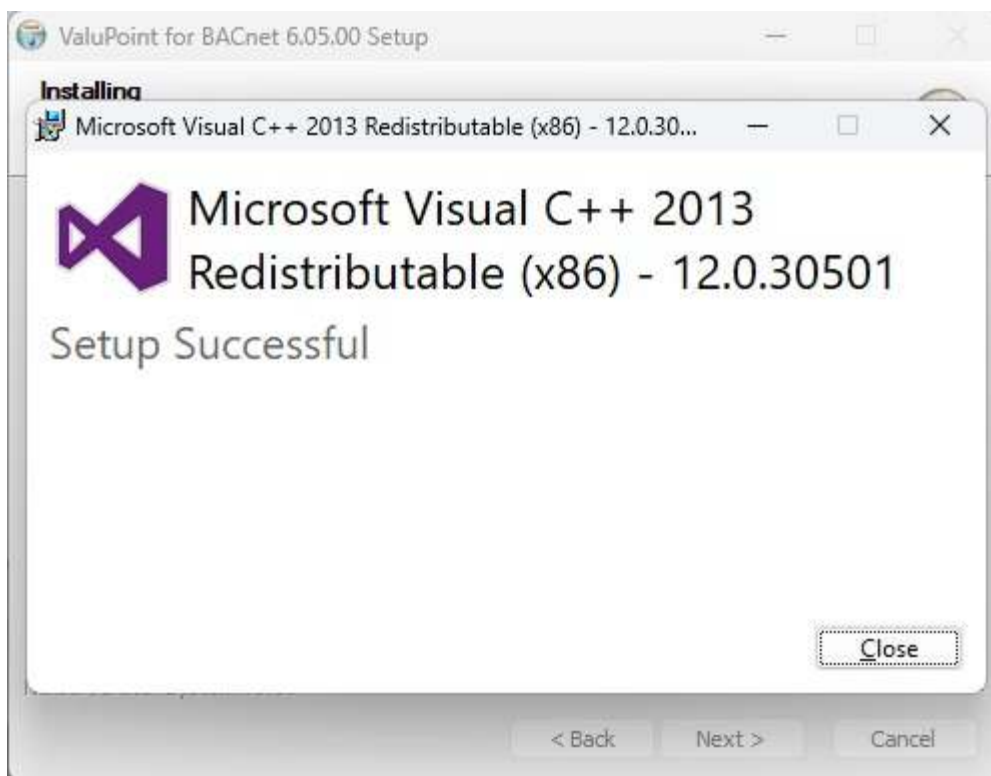
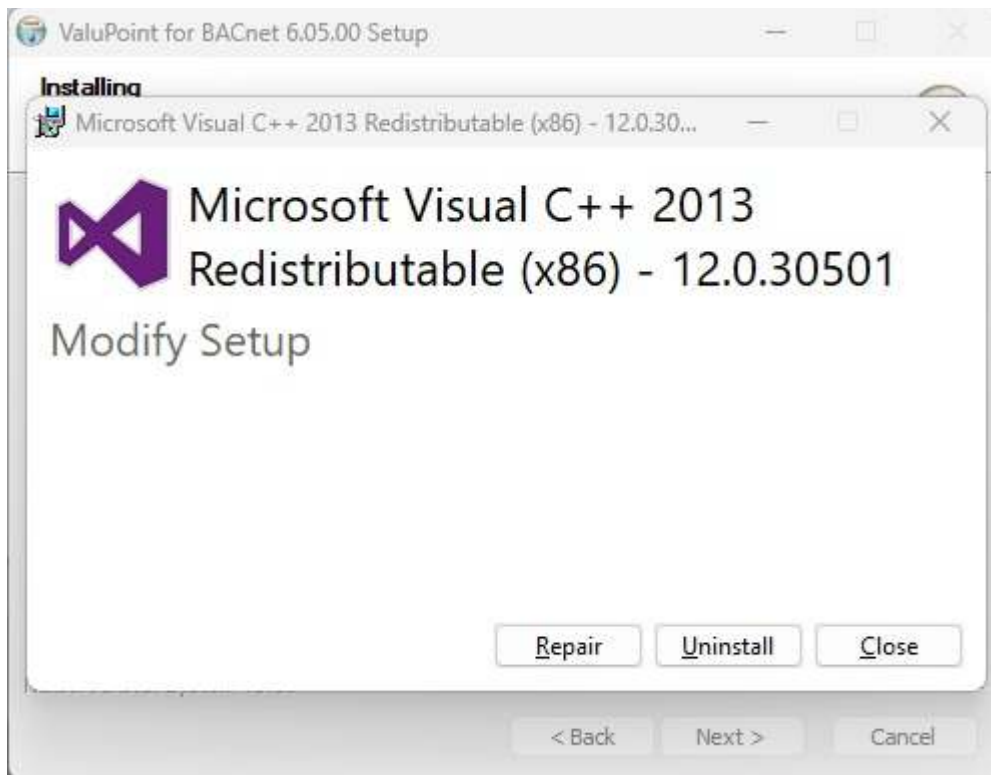
Double click the icon to run the setup.exe. You will be questioned about whether to continue because Windows cannot verify the publisher of the software. Permit installation to continue. The sequence of installer screens include the following on Windows 11:



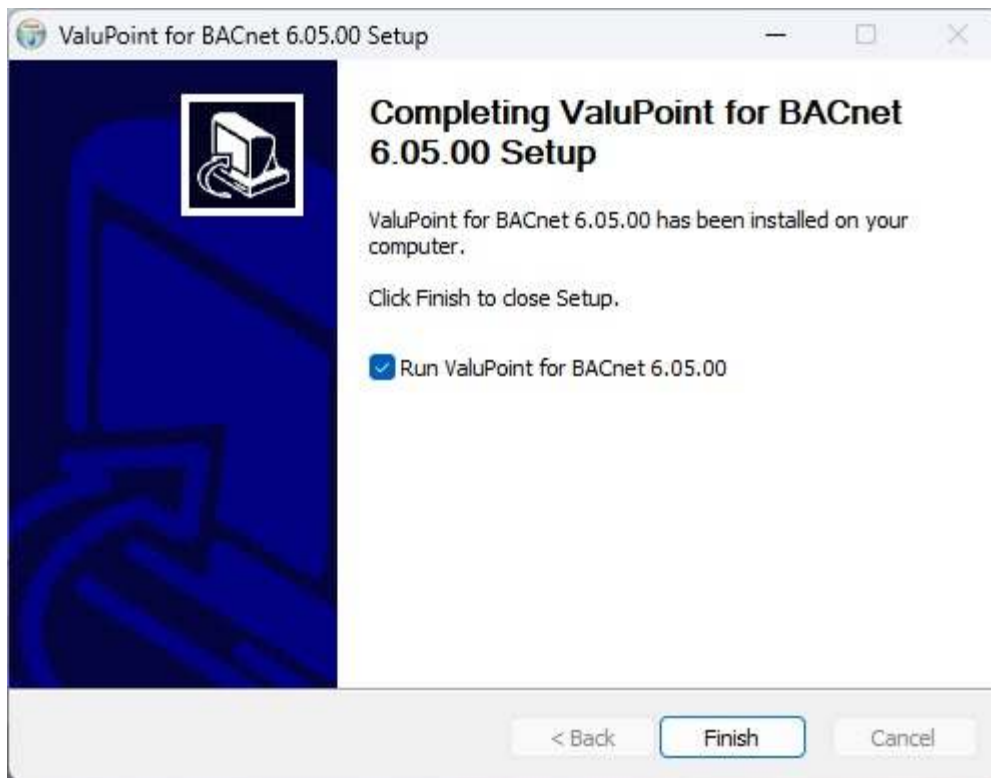




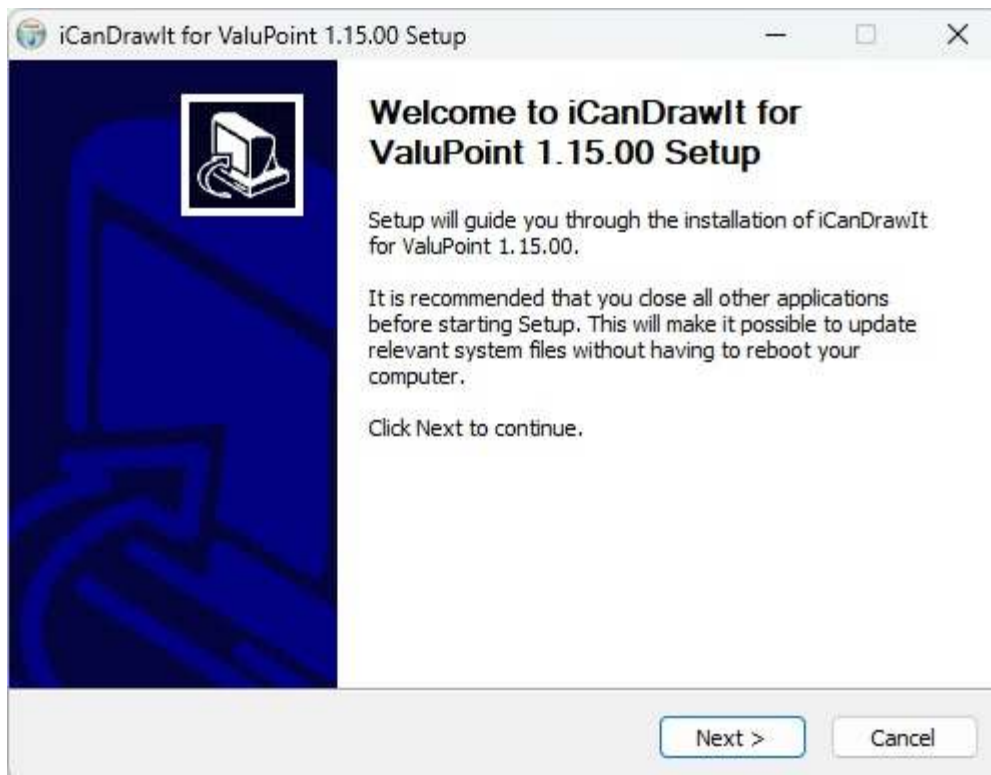
The installer will check to see whether Visual C++ support is already installed on your system, and install it if not. This is standard software provided by Microsoft. If it has already been installed previously, it will ask whether to repair or uninstall. Select Repair.



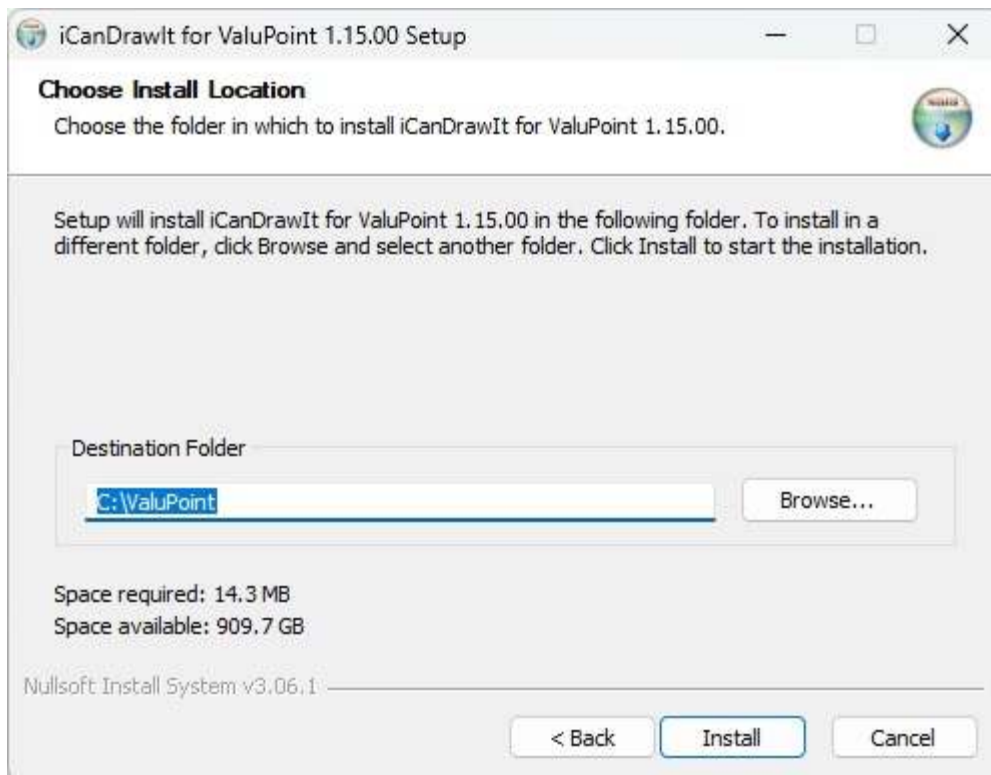
When you get to the "Finish" screen, you are ready to go.



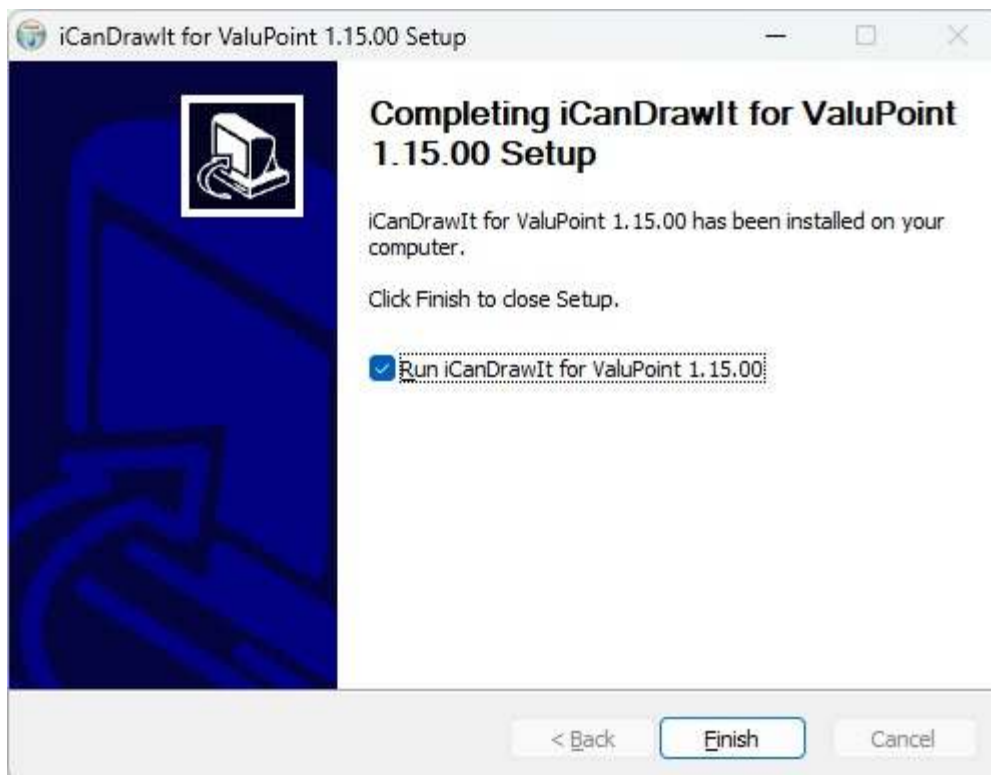
Next, proceed to install i.CanDrawIt. This part is optional. If you will not be using the ValuPoint as a programmable controller, you can skip this step. The first installer screen looks like this:



The installation directory should be the same directory that VP6-1430 was installed into.



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



## 2.2 Installing the USB to MS/TP adapter

USB installation is only required if you will be using the MTX002 to connect your PC to the configuration tool via MS/TP.

Start by installing the USB driver provided by Control Solutions. Complete instructions for USB driver installation, including step by step screen shots, are provided in Appendix F. Once the driver software is installed, install the USB adapter by plugging it into your PC. It will then proceed with the "installing driver software" process.

### 2.3 Connecting the ValuPoint

Connect power to the terminals marked POWER and GND. Connect the RS485 network to NET+, NET-, and SG. The serial communication signal ground marked SG is NOT electrically common to any other GND terminal on the device, and therefore must be connected to the network signal ground in order to complete the communication connection.



The VP6-1430 connects to the configuration and programming tools via the BACnet MS/TP port. You can connect via a BACnet IP to MS/TP router, or by using the Control Solutions MTX002 USB adapter (see following section).

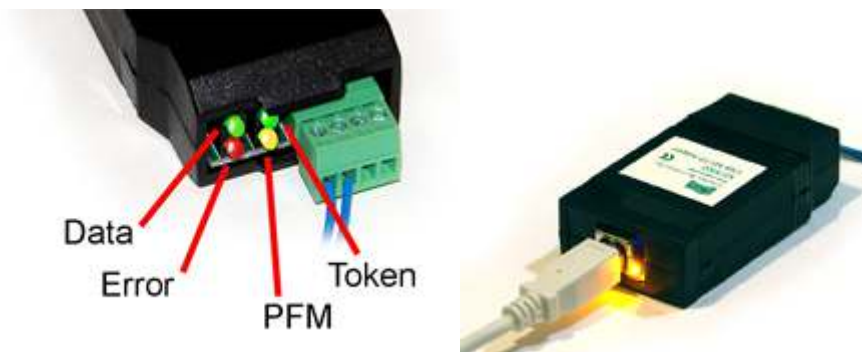
### 2.4 Connecting the USB Adapter

For MS/TP devices, 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.

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.

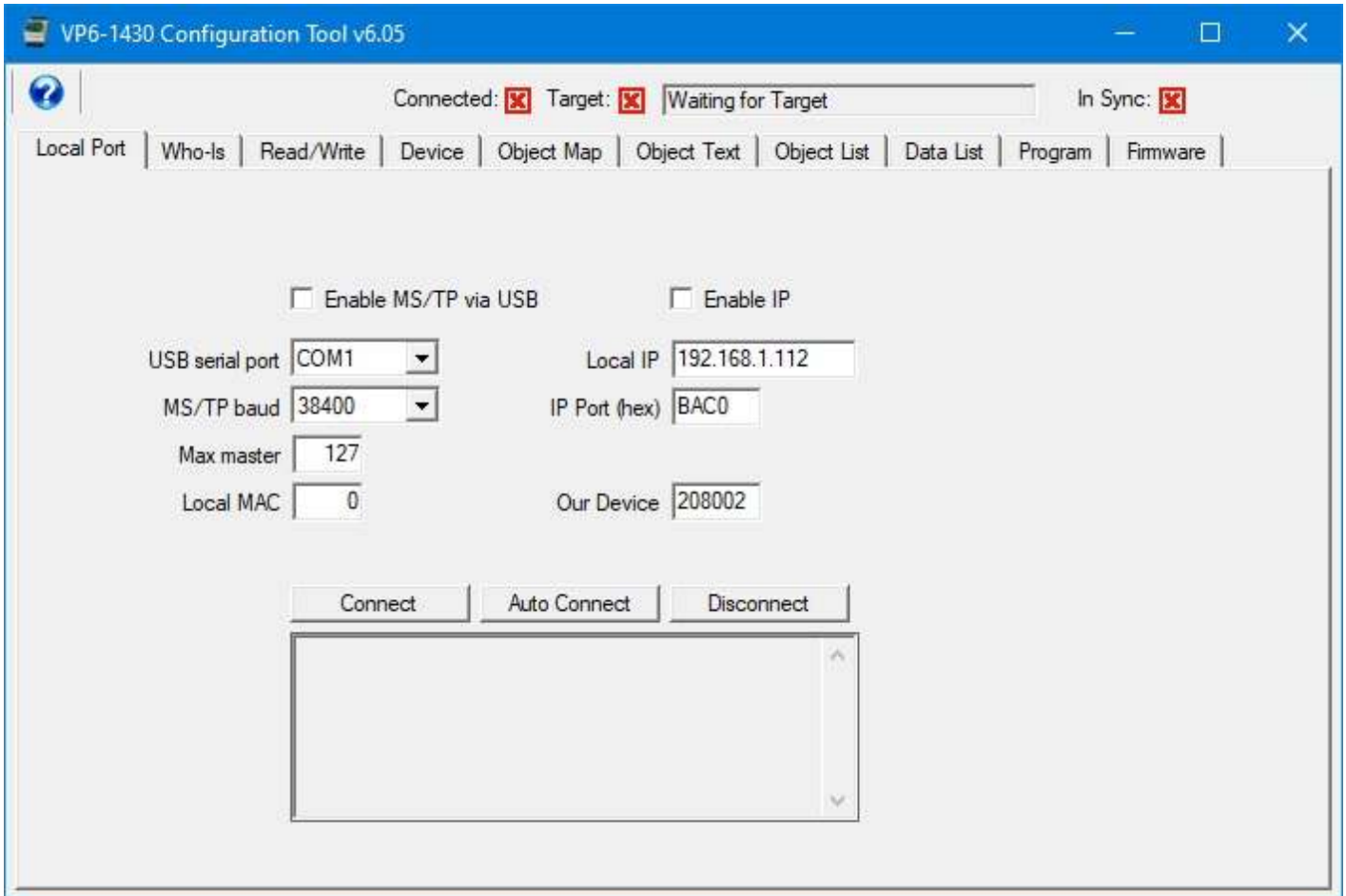
The PFM LED flashes yellow each time the adapter sends a Poll For Master (PFM).

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).



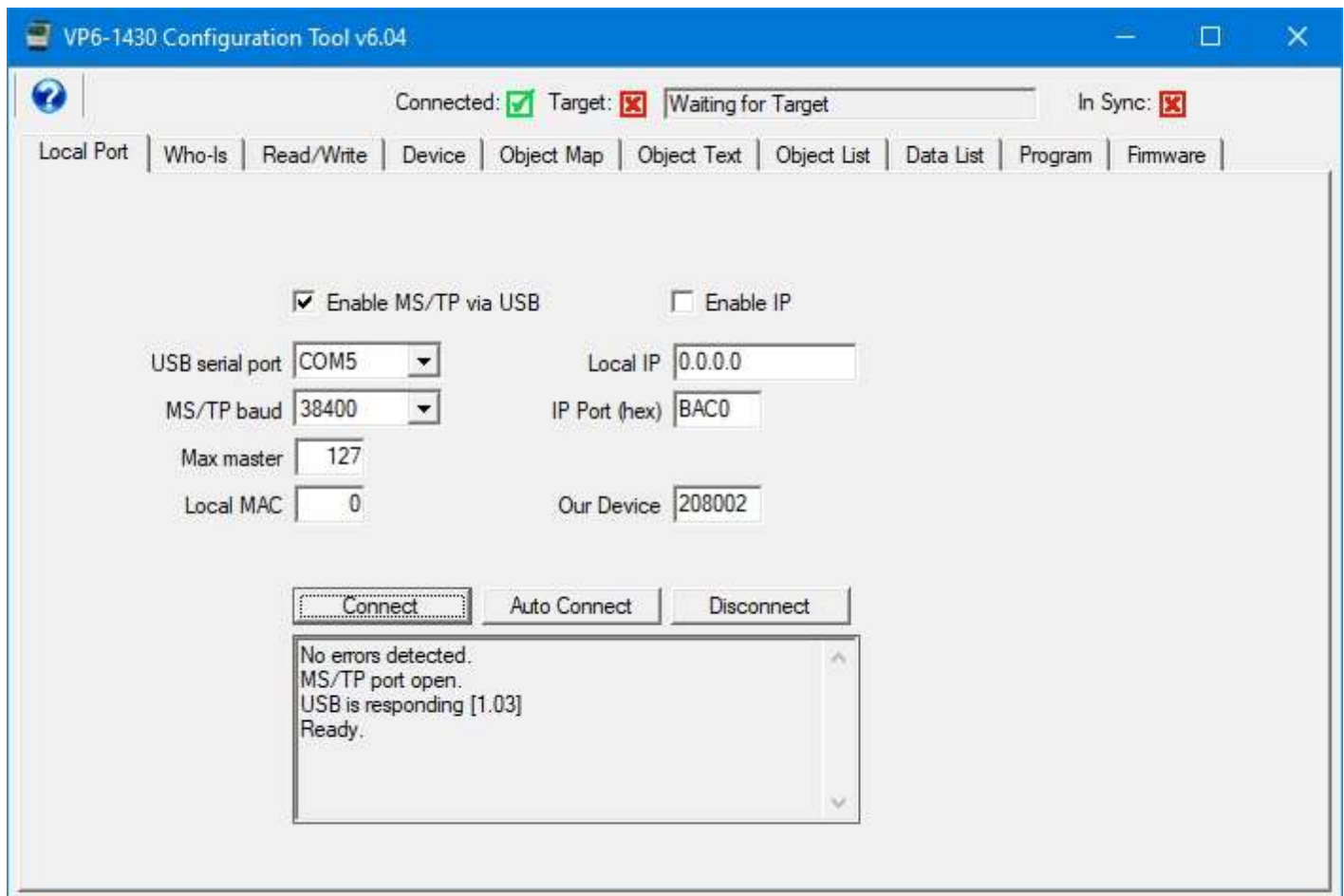
## 3 Local Port Page



The configuration tool will act as a BACnet device on the network. To connect the configuration tool to the BACnet network, you will use the Local Port page in the configuration tool. The configuration tool's Device instance will be 'Our Device'.

### 3.1 Connect with BACnet MS/TP

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. Otherwise, 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 208002 is already used by some other device, change the 'Our Device' number before connecting.

If the MS/TP MAC address displayed is already used by some other device, enter a different MAC address before clicking Connect. This MAC address will be used by the configuration tool. You should also enter a Max Master value other than the default 127 if you know some other value is in use.

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 instructions.

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



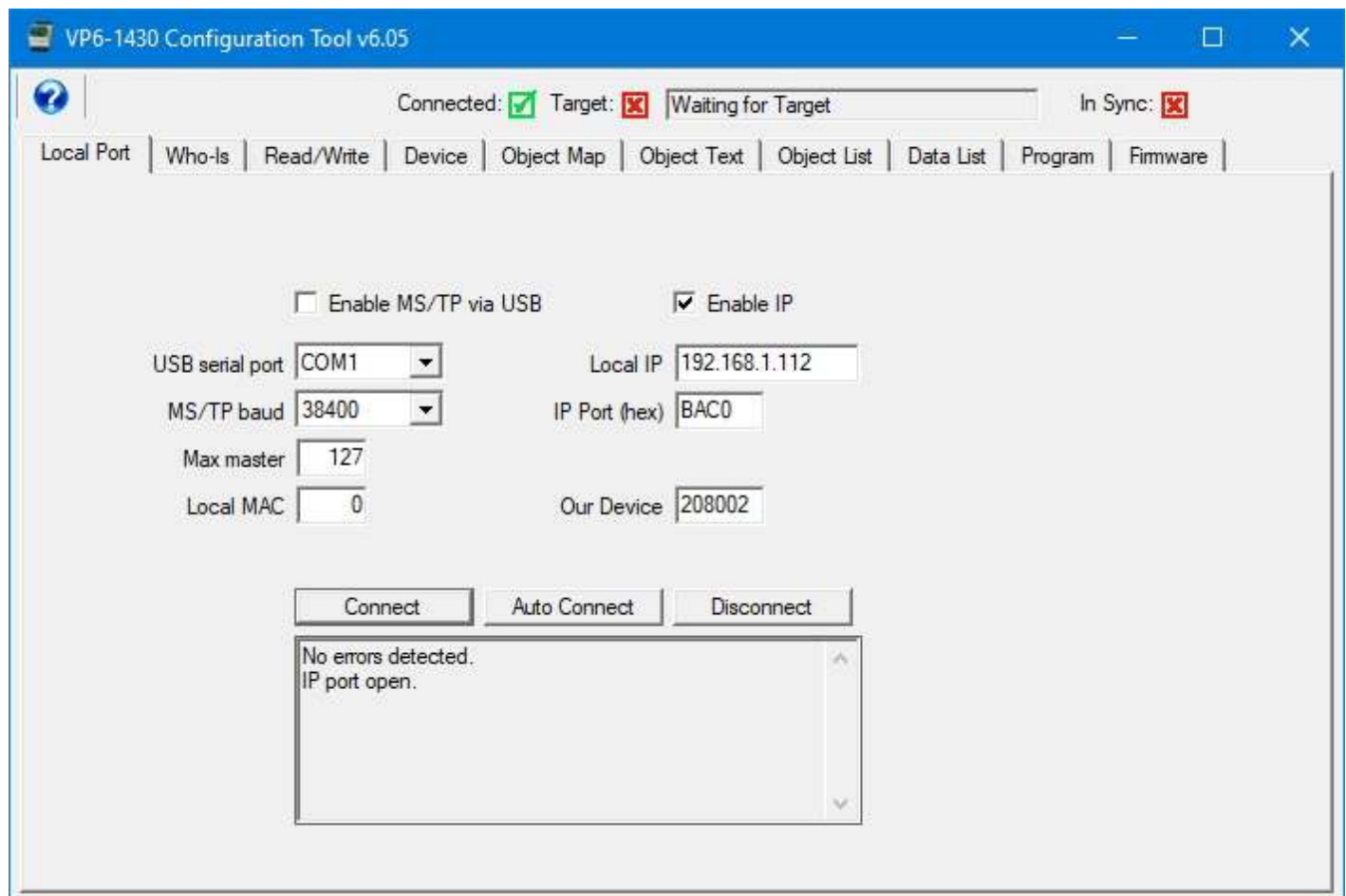
responding. Ready." then you are not ready.

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 Connect with BACnet IP

Check the 'Enable IP' box if you will be accessing the ValuPoint via BACnet IP. You can access the VP6-1430 using BACnet IP via an IP to MS/TP router. Enter the port number (0xBAC0 is the default) in hexadecimal. If device instance 208002 is already used by some other device, change the 'Our Device' number before connecting. The configuration tool itself becomes a BACnet device on the network, and therefore requires its own device instance.

Upon opening the program, the Local IP window will display the IP address that Windows has chosen as its default. However, if this is not the desired interface, enter the IP address of the interface you would rather use, then check Enable IP, and click Connect.

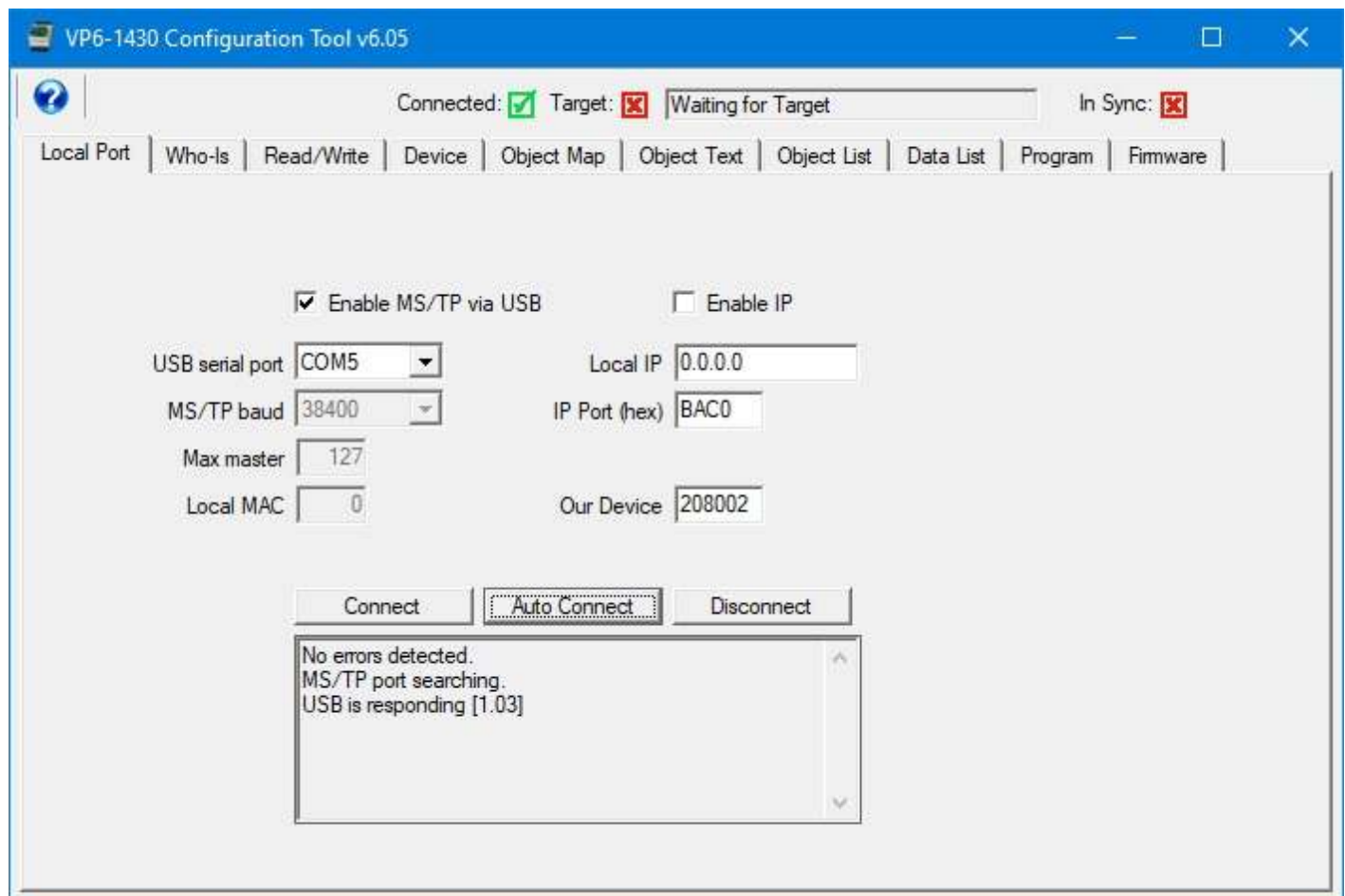


If the IP port is successfully connected, you will see the message "No errors detected. IP port open." You are now ready to move on to the Who-Is page.

### 3.3 Auto Connect (MS/TP)

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.

VP6-1430 Configuration Tool v6.05

Connected:  Target:  Waiting for Target In Sync:

Local Port | Who-Is | Read/Write | Device | Object Map | Object Text | Object List | Data List | Program | Firmware

Enable MS/TP via USB  Enable IP

USB serial port: COM5 Local IP: 0.0.0.0

MS/TP baud: 38400 IP Port (hex): BAC0

Max master: 127

Local MAC: 127 Our Device: 208002

Connect Auto Connect Disconnect

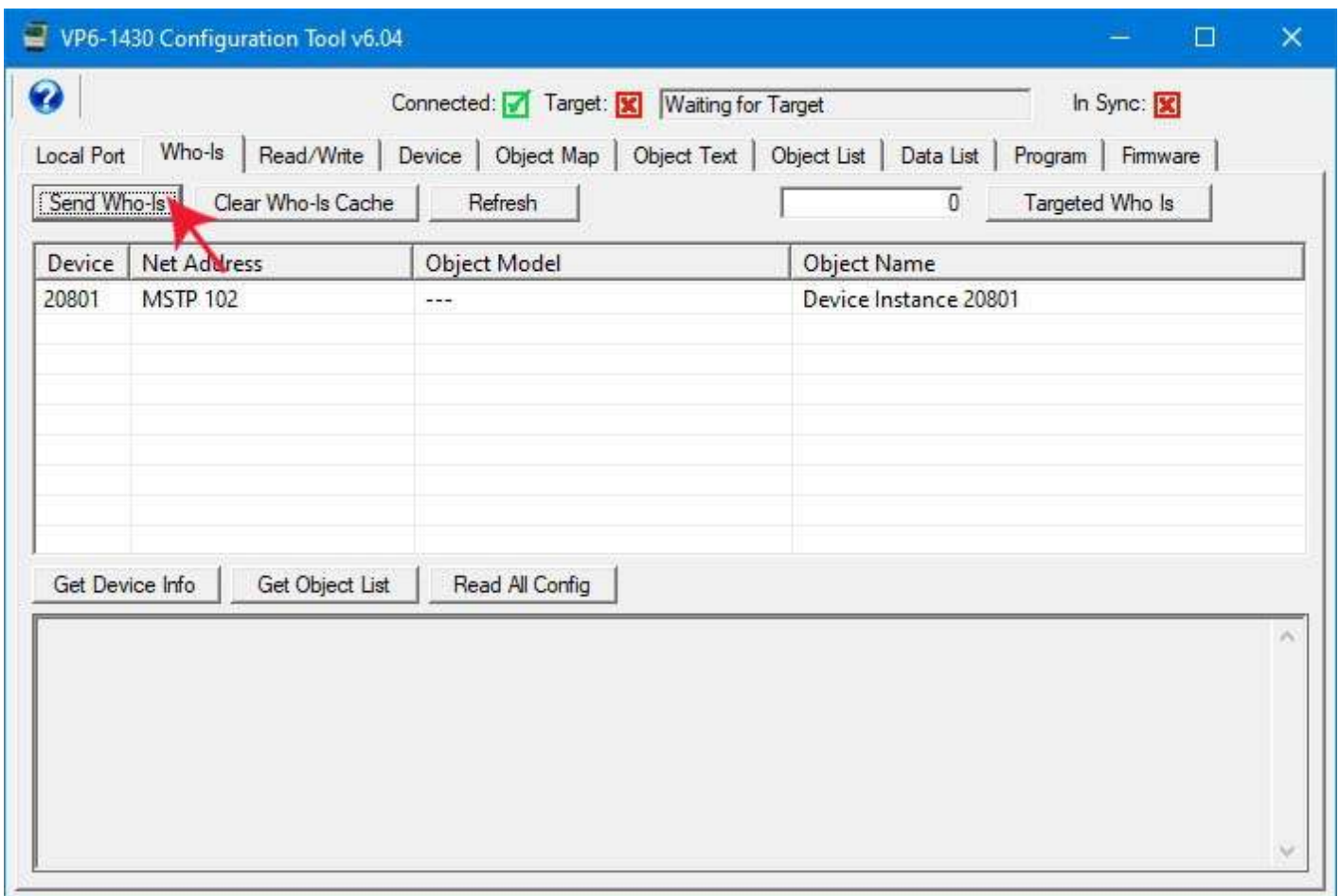
No errors detected.  
MS/TP port searching.  
USB is responding [1.03]  
Auto connected [1.03]



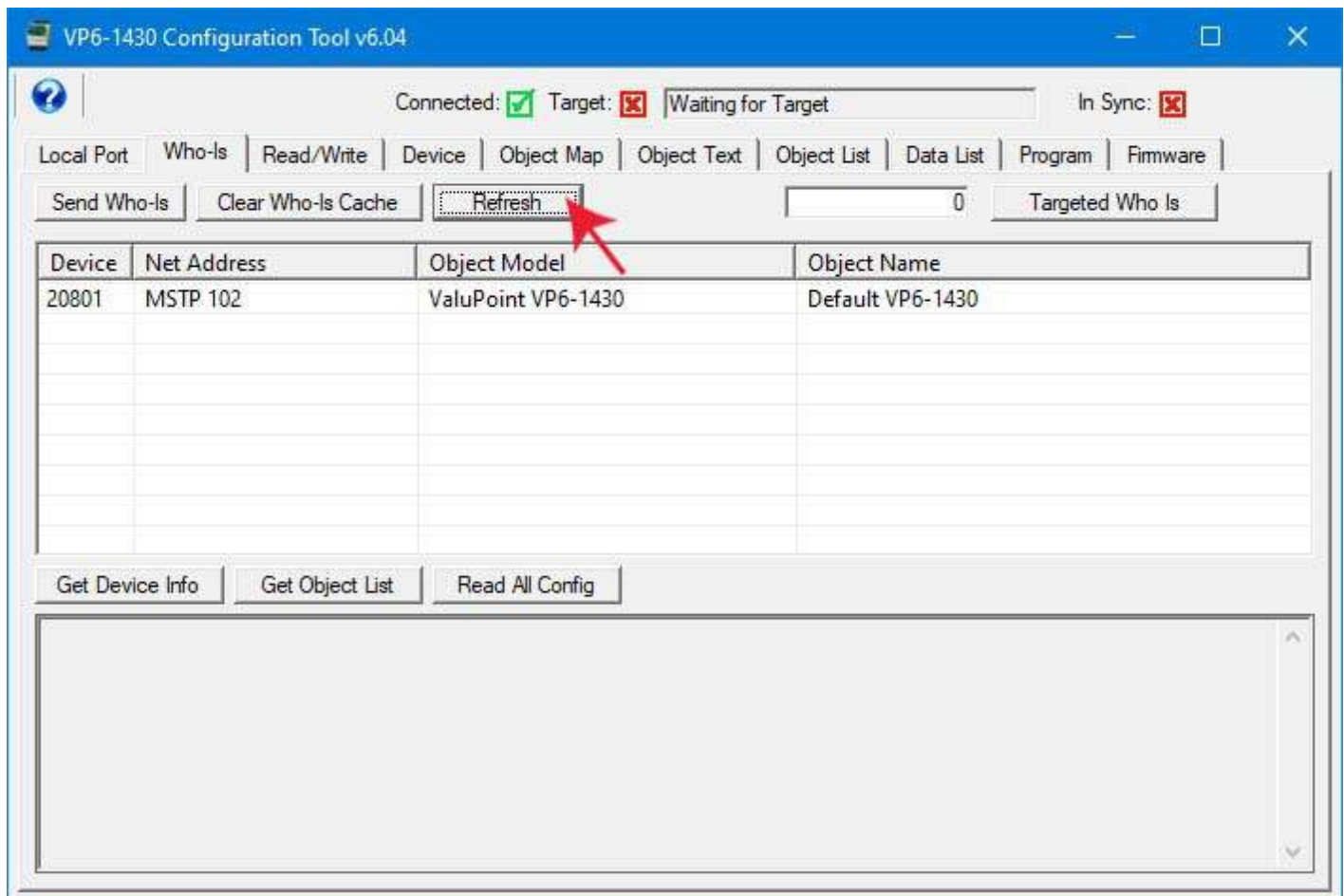
## 4 Who-Is Page

### 4.1 Finding Devices

Start by connecting the configuration tool to the network as outlined in section 3 of this user guide. Then come here to the Who-Is page. At first, there will be no devices listed. Click the "Send Who-Is" button to discover devices on the MS/TP network.

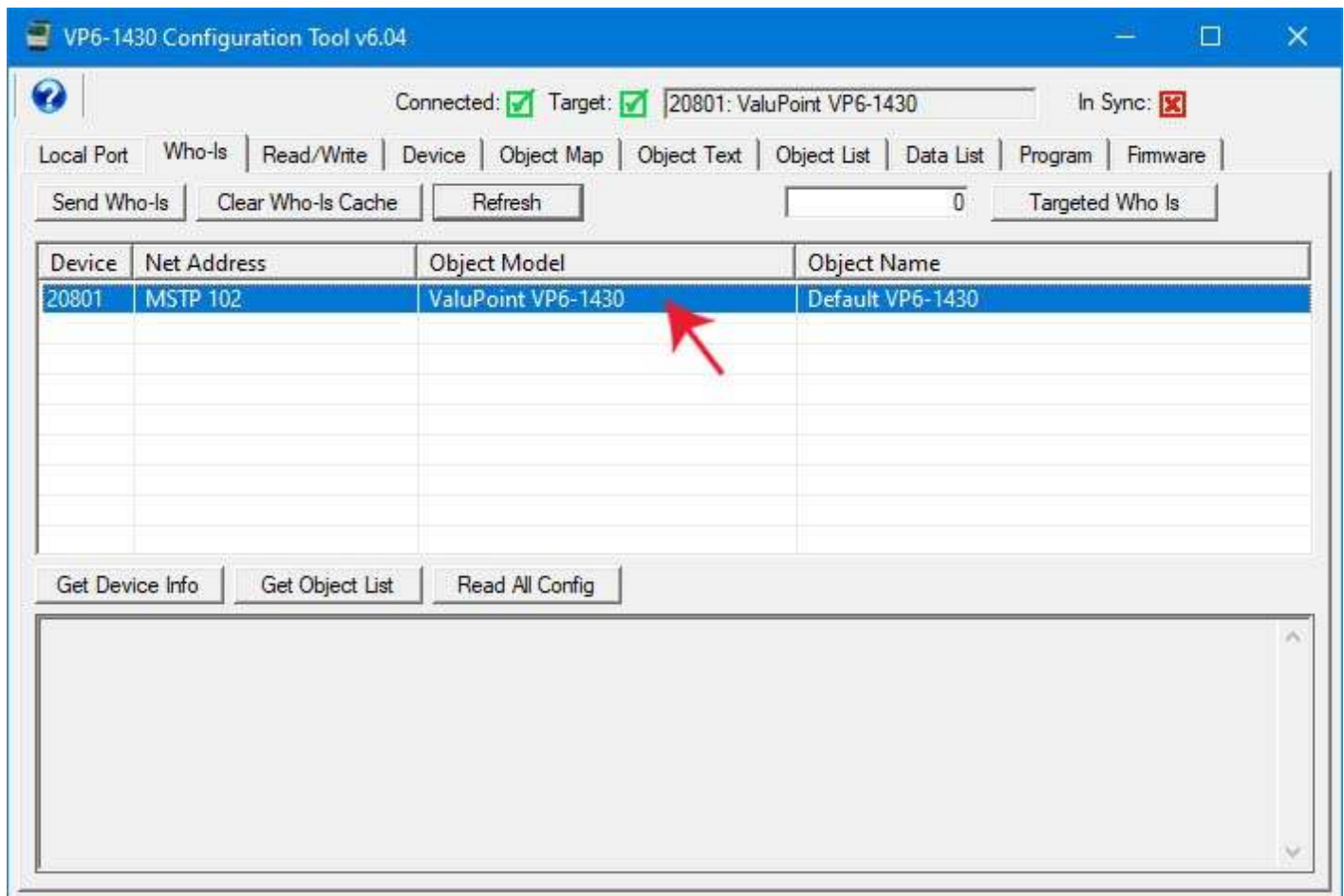


Click the Refresh button to cause the tool to request additional information from each device on the list, namely object model and object name as found in each device's BACnet Device object.



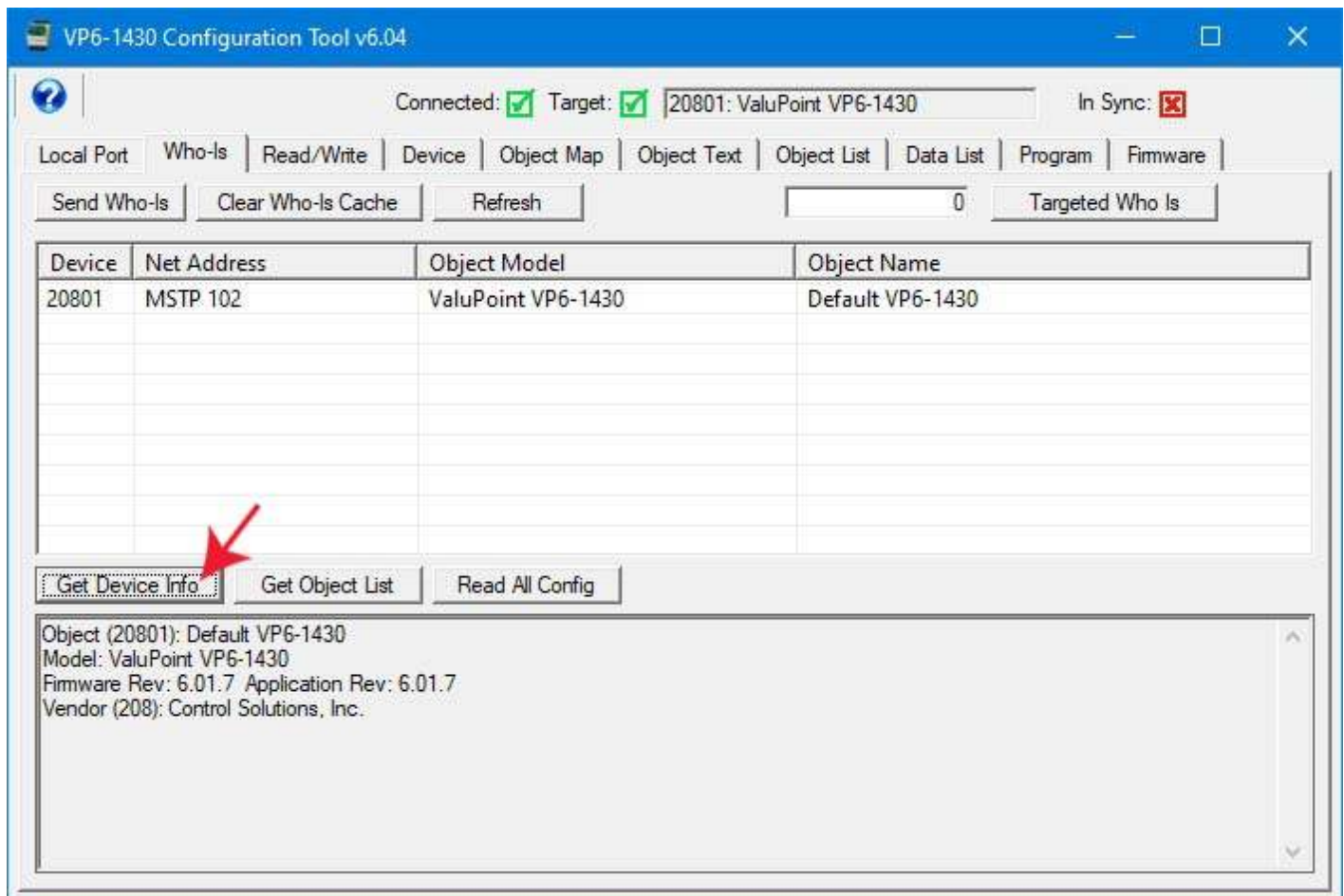
## 4.2 Select Target

In order to start interacting with a specific device, you need to double click that device on the list. Once selected, the icon to the right of "Target" will turn green, and the device instance and object model will appear in the target window at the top of the screen.

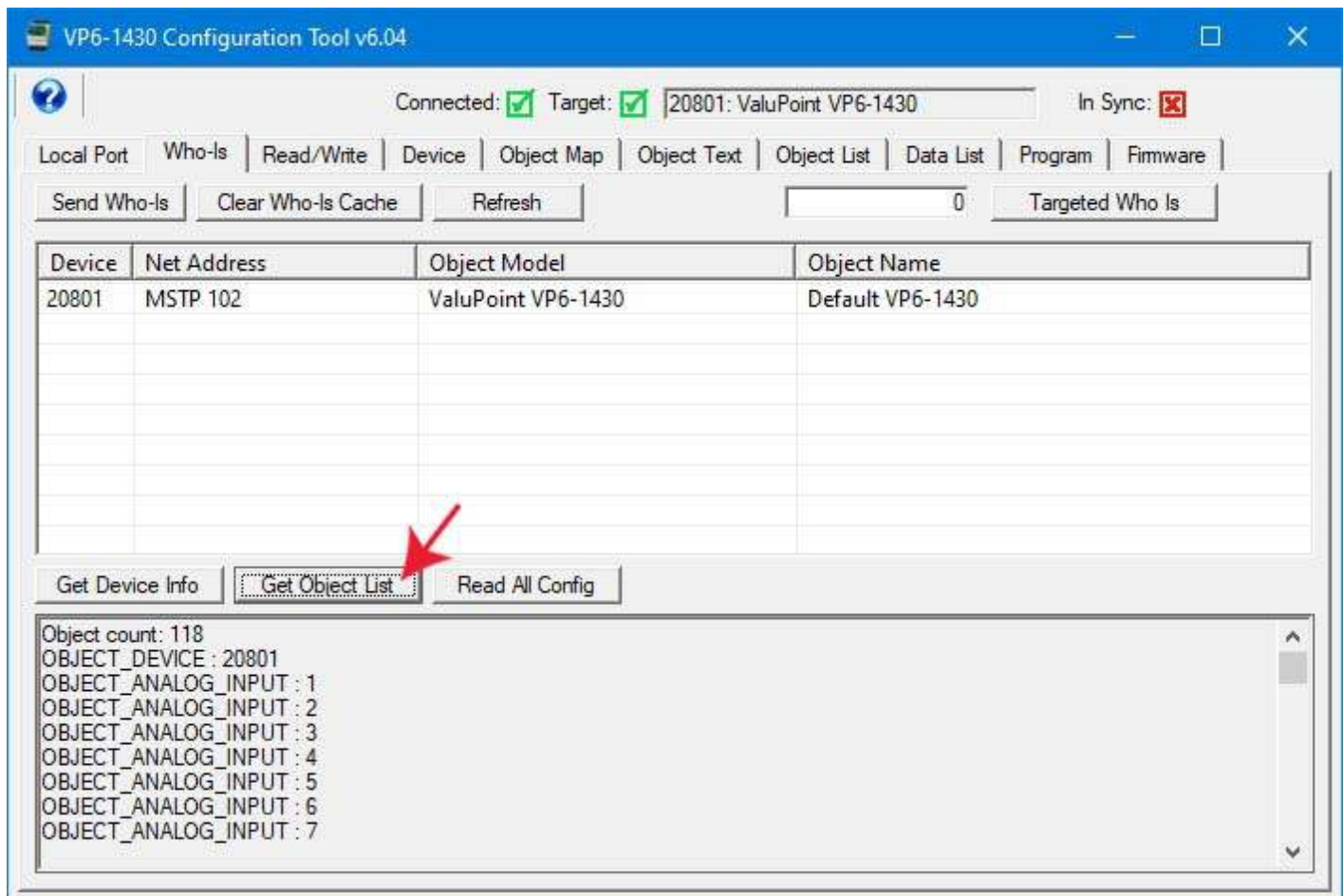


### 4.3 Getting Device Information

Click the Get Device Info button to get additional information about the selected device, including things like firmware revision. This step is optional and has no bearing on configuring the device.



You can also retrieve a listing of all objects found in the device by clicking the Get Object List button. Both of these buttons will work on any BACnet device regardless of whether it is a VP6-1430.



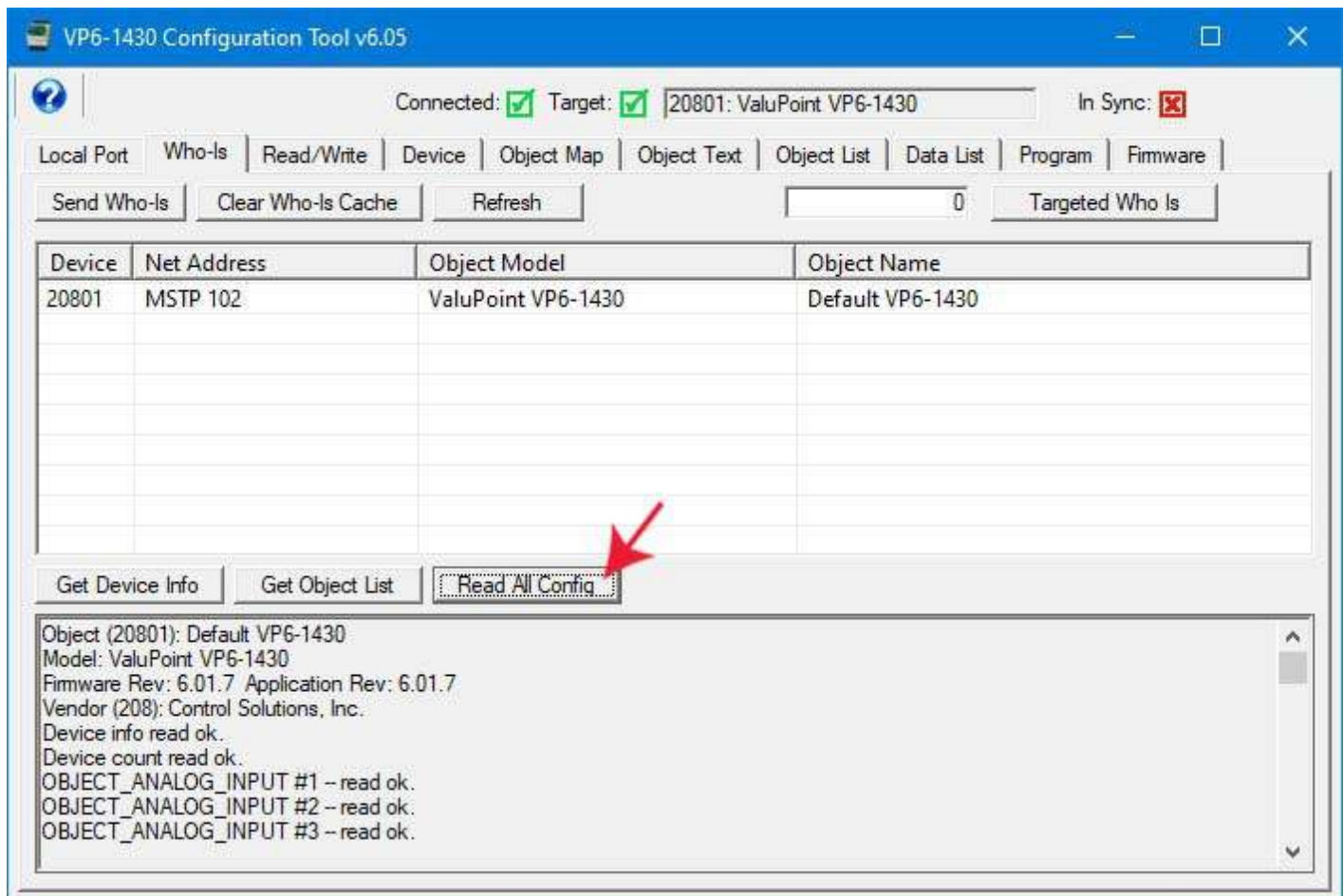
#### 4.4 Clear Who-Is Cache

If you want to start over on the device discovery process, click the Clear Who-Is Cache button, followed by Send Who-Is, and then Refresh after devices appear. **The only time this step becomes a requirement is if you change the device instance of the device you are in the process of configuring. If the device instance is changed, you must rediscover it under its new identity.**

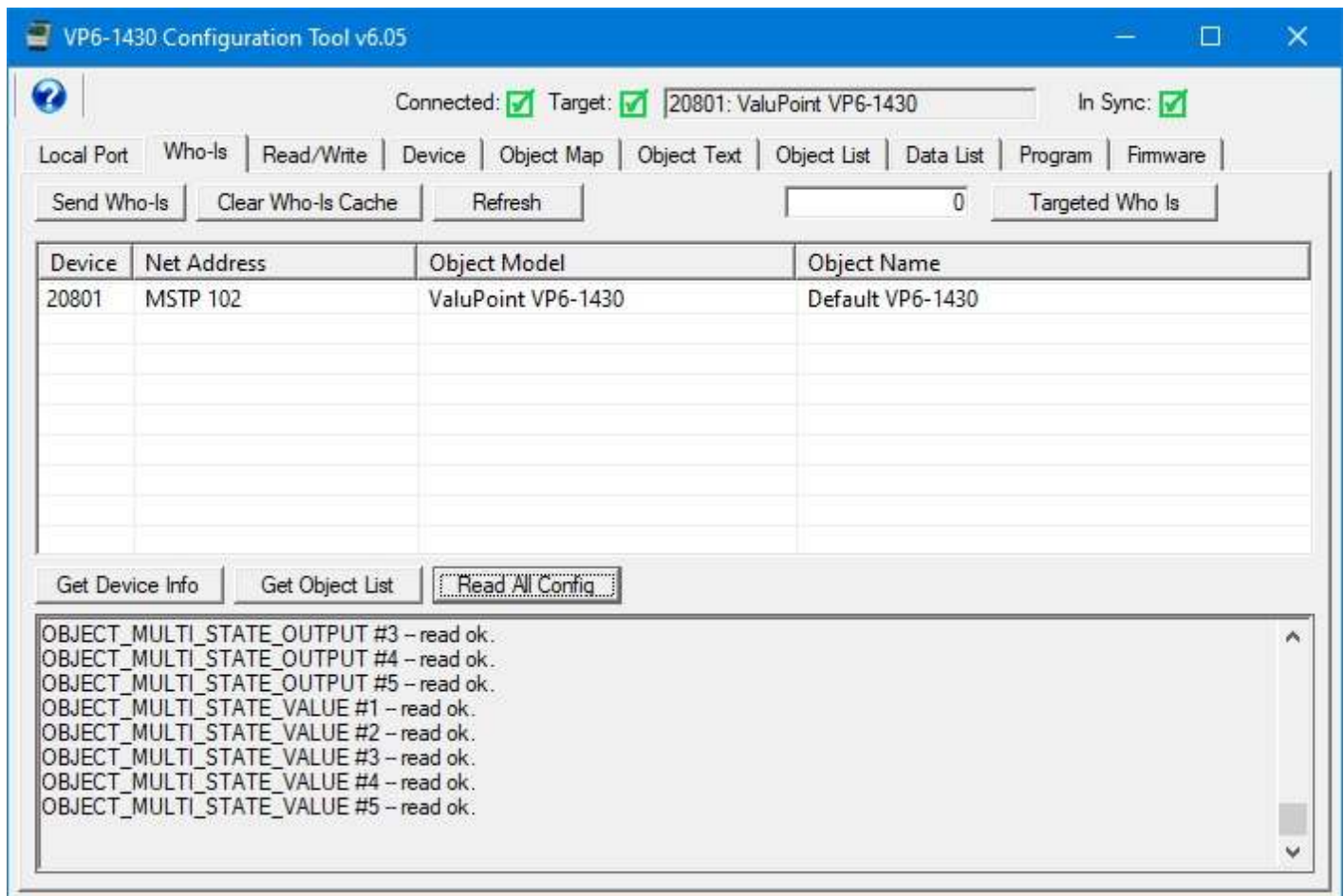
#### 4.5 Read All Configuration

Click the Read All Config button to retrieve all configuration information about the connected device. This button only applies to ValuPoint VP6-1430. Using this button on any other type of device will fail because it will not contain configuration properties in its objects. While configuration properties are being retrieved from the device, progress will be indicated in the status window at the bottom.





Once the "read all config" process is complete, the In Sync icon in the upper right corner will turn green. The "In Sync" icon that appears in the upper right corner of every page is the global "everything is in sync". If red, it means there is at least one red icon on some page somewhere in the configuration tool. A red icon (red X or red dot) means the configuration displayed on the screen in the configuration tool is not necessarily what is actually in the device. If the icon is green (green check mark or green dot), then the device is in sync with what is displayed.



## 4.6 Who-Is via IP Router

The ValuPoint configuration tool will work when connected via a BACnet IP router (check Enable IP on Connect page). The Who-Is list of MS/TP 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.

VP6-1430 Configuration Tool v6.05

Connected:  Target:  20801: ValuPoint VP6-1430 In Sync:

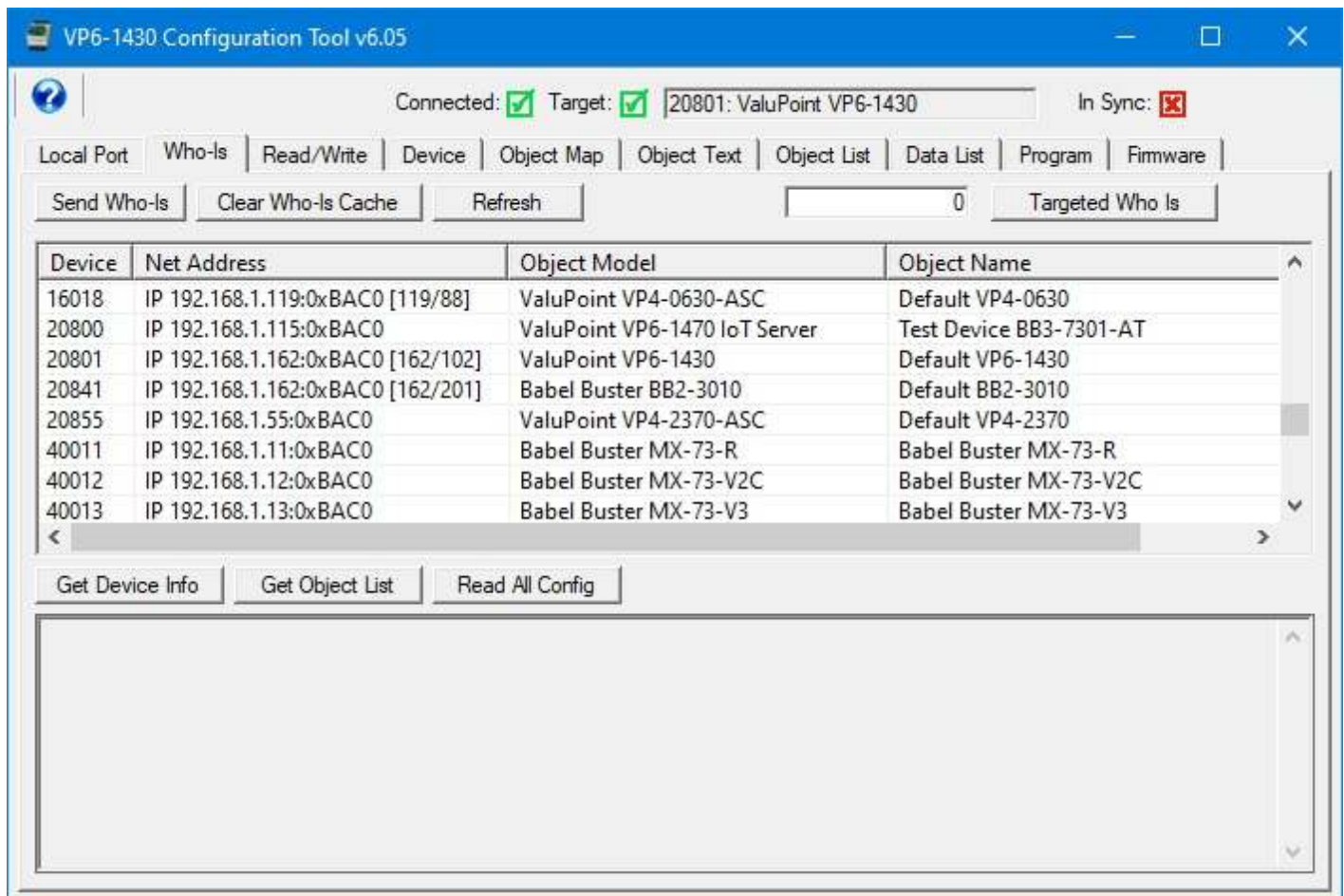
Local Port | Who-Is | Read/Write | Device | Object Map | Object Text | Object List | Data List | Program | Firmware

Send Who-Is | Clear Who-Is Cache | Refresh | 0 | Targeted Who Is

Device	Net Address	Object Model	Object Name
16018	IP 192.168.1.119:0xBAC0 ...	ValuPoint VP4-0630-ASC	Default VP4-0630
20800	IP 192.168.1.115:0xBAC0	ValuPoint VP6-1470 IoT Server	Test Device BB3-7301-AT
20801	IP 192.168.1.162:0xBAC0 ...	ValuPoint VP6-1430	Default VP6-1430
20841	IP 192.168.1.162:0xBAC0 ...	Babel Buster BB2-3010	Default BB2-3010
20855	IP 192.168.1.55:0xBAC0	ValuPoint VP4-2370-ASC	Default VP4-2370
40011	IP 192.168.1.11:0xBAC0	Babel Buster MX-73-R	Babel Buster MX-73-R
40012	IP 192.168.1.12:0xBAC0	Babel Buster MX-73-V2C	Babel Buster MX-73-V2C
40013	IP 192.168.1.13:0xBAC0	Babel Buster MX-73-V3	Babel Buster MX-73-V3

Get Device Info | Get Object List | Read All Config

To see the full network address in cases like this, you need to expand the address column. Here we can see several BACnet IP devices accessible via the BACnet router.



## 4.7 Targeted Who-Is

It is possible on a large network, especially when connected via IP, that there will be so many I-Am replies coming back in response to the Who-Is request that they do not all get through. It has been verified using WireShark that it is not a matter of the tool missing them, it is a matter of the I-Am simply not arriving at all because the network is flooded. The test scenario proving this involved multiple routers, multiple networks, and many devices. You can re-click the Send Who-Is button to try to capture the missing devices, but the same flood of I-Am replies is likely to happen.

To send the Who-Is explicitly to only your device of interest, enter the desired device instance in the window next to the Targeted Who Is button, and click that button. Only that device will now respond with its I-Am reply. Then click the Refresh button to obtain the model and object name.

VP6-1430 Configuration Tool v6.05

Connected:  Target:  Waiting for Target In Sync:

Local Port | Who-Is | Read/Write | Device | Object Map | Object Text | Object List | Data List | Program | Firmware

Send Who-Is | Clear Who-Is Cache | Refresh | 20801 | Targeted Who Is

Device	Net Address	Object Model	Object Name
20801	IP 192.168.1.162:0xBAC0 [162/102]	ValuPoint VP6-1430	Default VP6-1430

Get Device Info | Get Object List | Read All Config



## 5 Read/Write Page

This page may be used as soon as a device is selected on the Who-Is page. The device does not have to be a VP6-1430 in order to use the generic object property read/write on this page. This can be a useful diagnostic for any BACnet device.

### 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'.

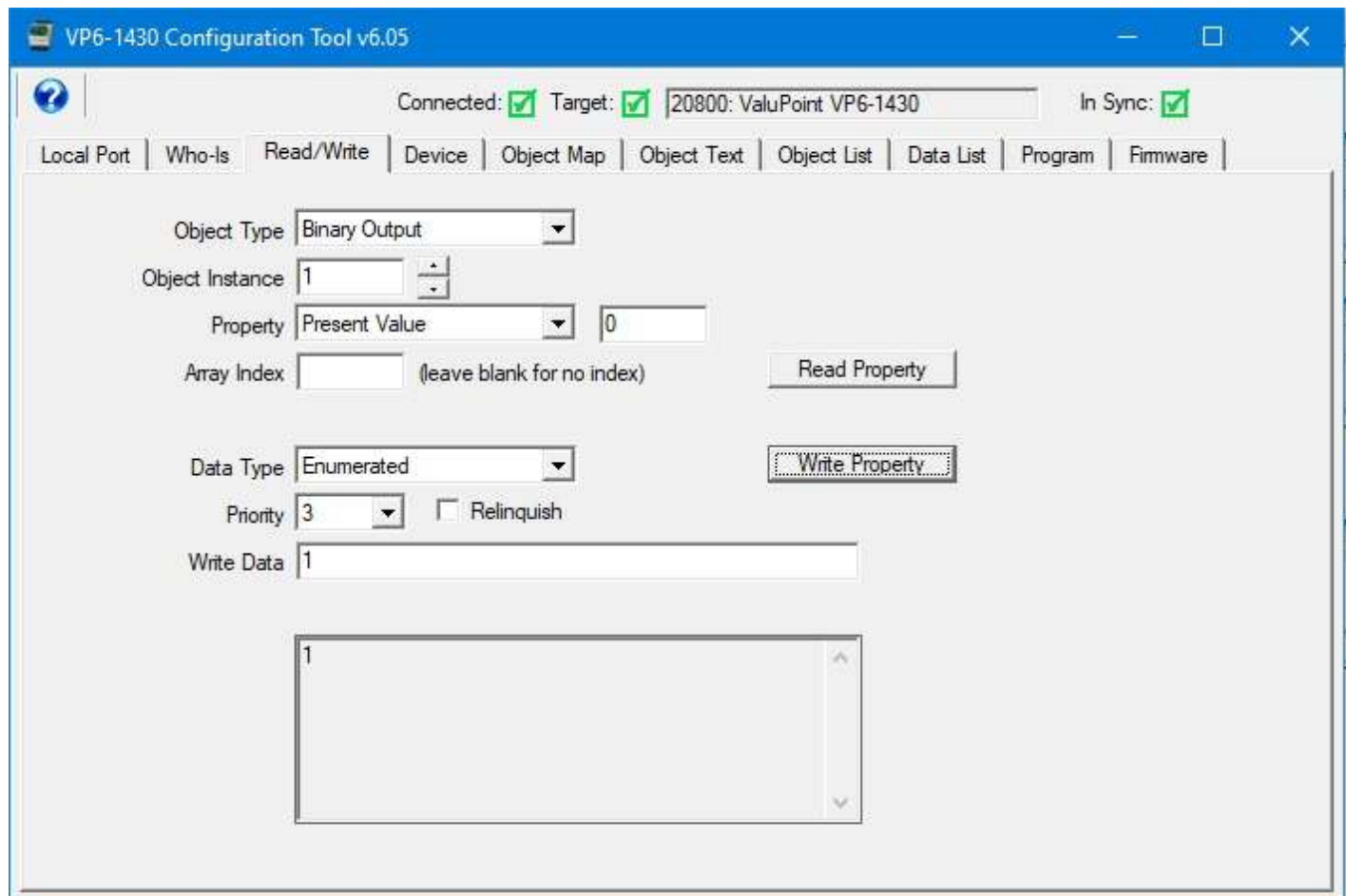
The screenshot shows the 'VP6-1430 Configuration Tool v6.05' interface. The 'Read/Write' tab is active. The 'Connected' and 'Target' checkboxes are checked, with the target address '20800: ValuPoint VP6-1430' and 'In Sync' checked. The 'Object Type' is set to 'Analog Input', 'Object Instance' is '6', 'Property' is 'Present Value', and 'Array Index' is blank. The 'Data Type' is 'Null', 'Priority' is 'None', and 'Relinquish' is unchecked. The 'Write Data' field is empty. The 'Read Property' button is highlighted. A log window at the bottom shows the value '20.000000'.

## 5.2 Write Property

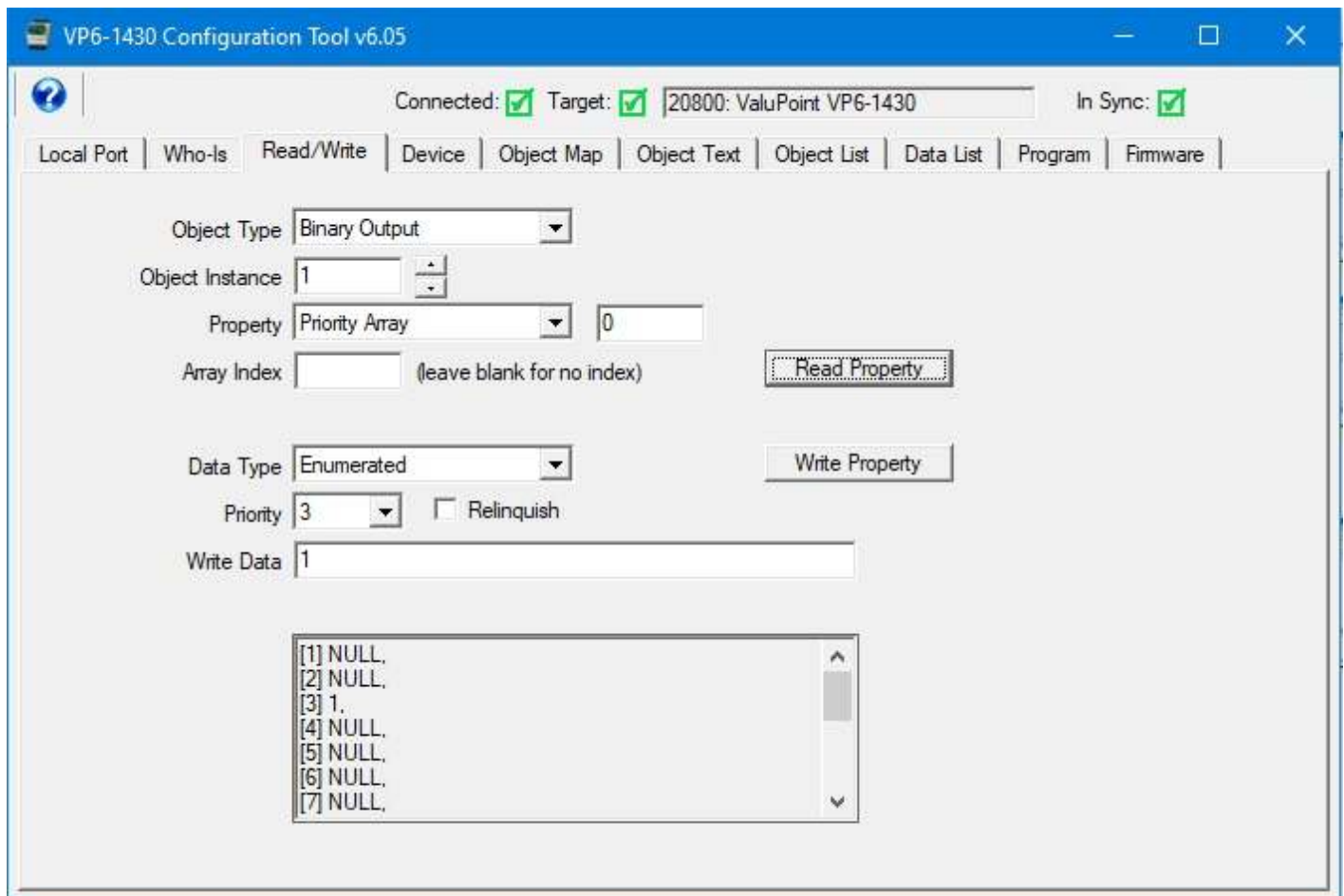
Select all of the same settings as you would for Read Property, and in addition, specify data type, priority (if commandable, use 'none' if not commandable), and data to write. Then click 'Write Property'.

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 2 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.

The following is an example of writing a commandable Present Value.



The resulting data priority array can be read as illustrated below.



To relinquish the commanded value, select the priority, and check Relinquish. Then click Write Property.



VP6-1430 Configuration Tool v6.05

Connected:  Target:  20800: ValuPoint VP6-1430 In Sync:

Local Port | Who-Is | Read/Write | Device | Object Map | Object Text | Object List | Data List | Program | Firmware

Object Type: Binary Output

Object Instance: 1

Property: Present Value 0

Array Index: (leave blank for no index)

Data Type: Enumerated

Priority: 3  Relinquish

Write Data: 1

NULL



## 6 Device Page

The "Device" page is where you create the VP6-1430's identity as a BACnet device.

### 6.1 Retrieve Device Object Information from Device

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).

Object Allocation	Present	Pending
Analog Input Count	0	0
Analog Output Count	0	0
Analog Value Count	0	0
Binary Input Count	0	0
Binary Output Count	0	0
Binary Value Count	0	0
Multistate Input Count	0	0
Multistate Output Count	0	0
Multistate Value Count	0	0
Multistate Max State	0	0

Object ID

Object ID is the device instance of this device. It must be unique on the network. When a BACnet client broadcasts the "Who-Is" request, this number will be sent back as this device's identity. If more than one device respond with the same device instance, then the client will be unable to communicate correctly with the devices.

Device Name	This Device Name must be a unique name on your BACnet network. When a BACnet client broadcasts a "Who-Has" request asking for a device object by a given name, you will create problems on the network if more than one device responds with the same device name. Therefore, be careful to see that the name is unique.
Device Description	The Device Description property of the Device object is there for your own use in whatever way you see fit, and does not need to be unique.
Device Location	The Device Location property of the Device object is there for your own use in whatever way you see fit, and does not need to be unique.
Auto-Reset Errors	<p>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.</p> <p>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.</p> <p>Many systems will report an object as "offline" because its fault status bit is set. It is not actually offline, and is in fact communicating just fine trying to tell you that there is a problem. But many front end systems don't recognize this and blindly claim "offline" as a result of the fault bit in the status property. If you are having this issue as the result of BACnet client communication errors, try setting the Auto-Reset Errors option here.</p>
Move Outputs to Binary Values	The relay outputs are assigned to Binary Output objects by default. If you would rather have them assigned to Binary Value objects, check this box.
Max Master	The Max Master determines the highest device MAC address that the device (or any MS/TP device) will try to poll for. It will default to 127, and you should leave this set to 127 unless you know exactly why you are changing it.
MS/TP Baud Rate	MS/TP Baud Rate needs to be set to match whatever speed your MS/TP network is running at. The VP6-1430 will support 9600, 19200, 38400, and 76800.

MS/ TP Station or MAC Addr	This is the MAC address of the device. The MAC address must be unique on the network. If more than one device is given the same MAC address, the MS/TP network will fail to function. Not only will this device fail to communicate, but the entire network will be corrupted by duplicate MAC addresses.
New Password	Use the New Password field if you want to change the default password to something else. Leave this field blank if you do not want to change the password. Any password in this window will take effect upon clicking the Change button.
Old Password	The Old Password is "buster" by default as shipped from the factory. You only need to use it when reinitializing the device.

## 6.2 Change Device Object Configuration

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 Device Instance).

To cause changes such as baud rate, MAC address, or Max Master setting to take effect, it is necessary to reinitialize the device. The new parameters are saved, but not put into use until you reinitialize the device (enter Old Password and click Reinit Device button) or power cycle the device. This reinitialization is required so that you don't immediately lose contact with the device when changing baud rate or MAC address. When you change port parameters like baud rate, you will also need to restart to configuration tool at the different baud rate.

VP6-1430 Configuration Tool v6.05

Connected:  Target:  20801: ValuPoint VP6-1430 In Sync:

Local Port | Who-Is | Read/Write | Device | Object Map | Object Text | Object List | Data List | Program | Firmware

Object ID (device instance) 20801 Reassign

Device Name  
Default VP6-1430

Device Description  
BACnet MS/TP Programmable I/O

Device Location  
Unassigned

Auto-Reset Errors  Move Outputs to Binary Values

Max Master 127 Get Info

MS/TP Baud Rate 38400 Change

MS/TP Station/MAC Addr 102

New Password

Old Password

Reint Device

Reconfigure Get Counts Check Fit

Device info written ok.

Object Allocation	Present	Pending
Analog Input Count	0	0
Analog Output Count	0	0
Analog Value Count	0	0
Binary Input Count	0	0
Binary Output Count	0	0
Binary Value Count	0	0
Multistate Input Count	0	0
Multistate Output Count	0	0
Multistate Value Count	0	0
Multistate Max State	0	0

### 6.3 Reassign Device Instance

To change the Object ID (device instance) property, enter the new object ID (device instance) and click the 'Reassign' button.

VP6-1430 Configuration Tool v6.05

Connected:  Target:  20801: ValuPoint VP6-1430 In Sync:

Local Port | Who-Is | Read/Write | Device | Object Map | Object Text | Object List | Data List | Program | Firmware

Object ID (device instance) 20802

Device Name  
Default VP6-1430

Device Description  
BACnet MS/TP Programmable I/O

Device Location  
Unassigned

Auto-Reset Errors  Move Outputs to Binary Values

Max Master 127

MS/TP Baud Rate 38400

MS/TP Station/MAC Addr 102

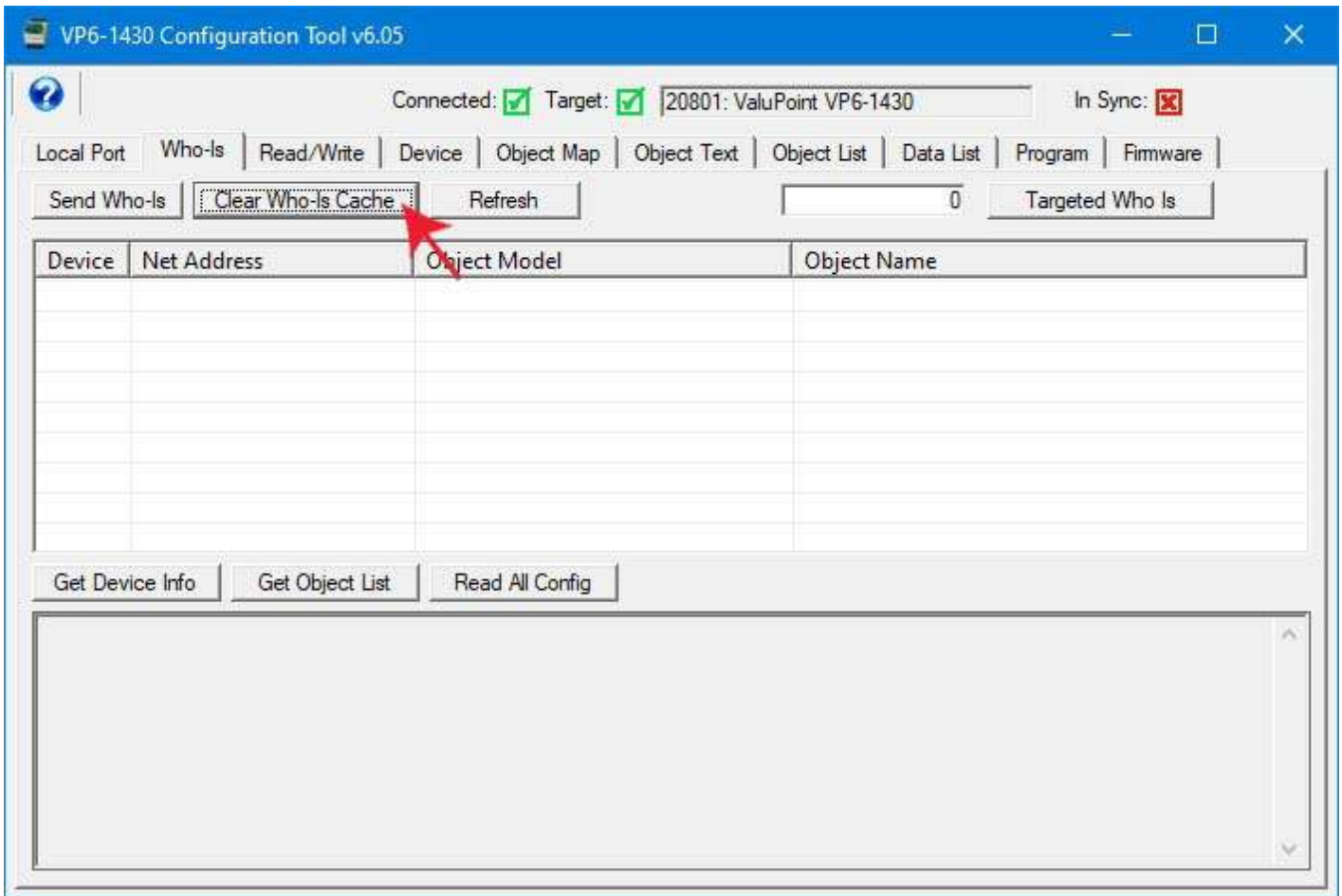
New Password

Old Password

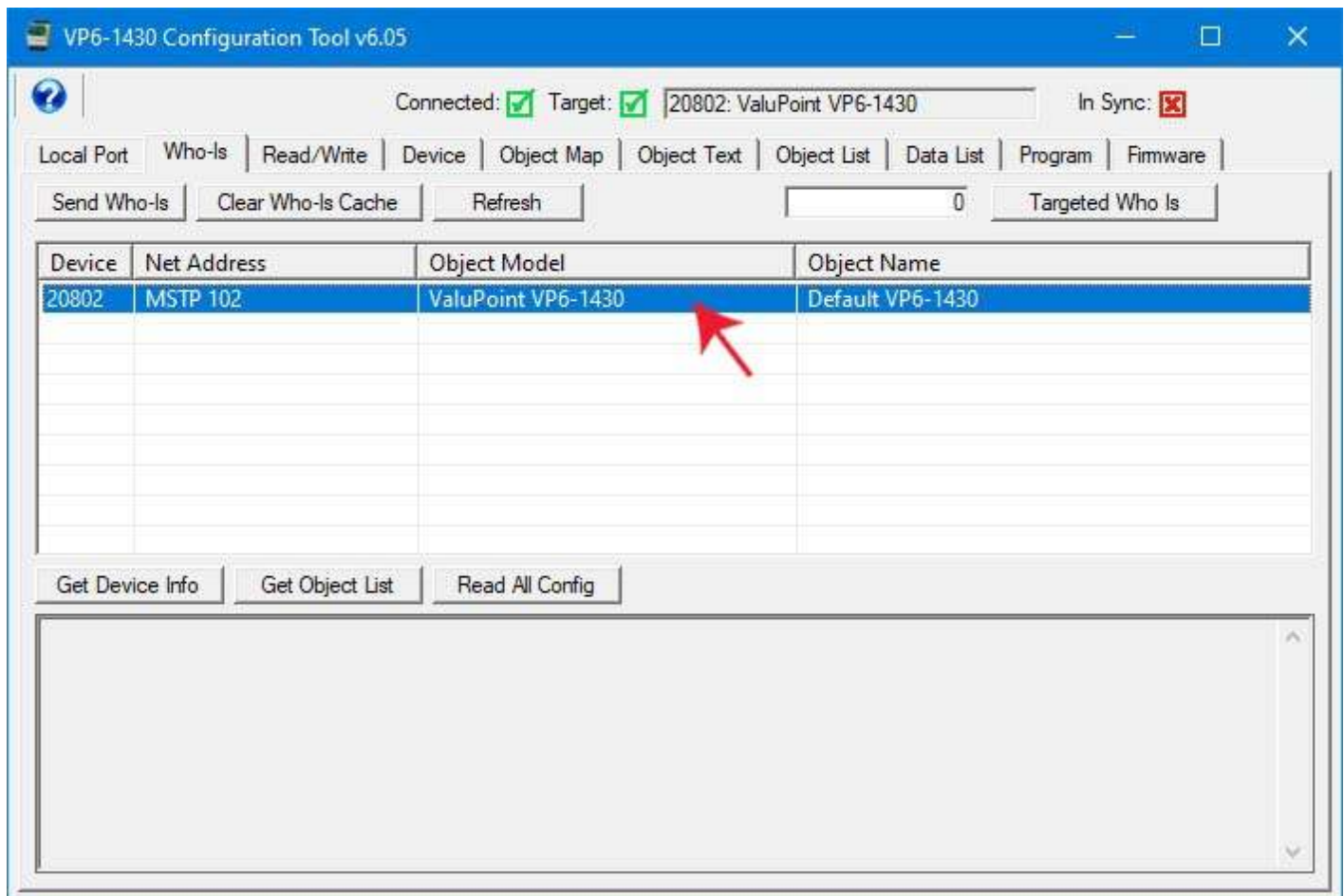
Object Allocation	Present	Pending
Analog Input Count	0	0
Analog Output Count	0	0
Analog Value Count	0	0
Binary Input Count	0	0
Binary Output Count	0	0
Binary Value Count	0	0
Multistate Input Count	0	0
Multistate Output Count	0	0
Multistate Value Count	0	0
Multistate Max State	0	0

Device info written ok.

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.

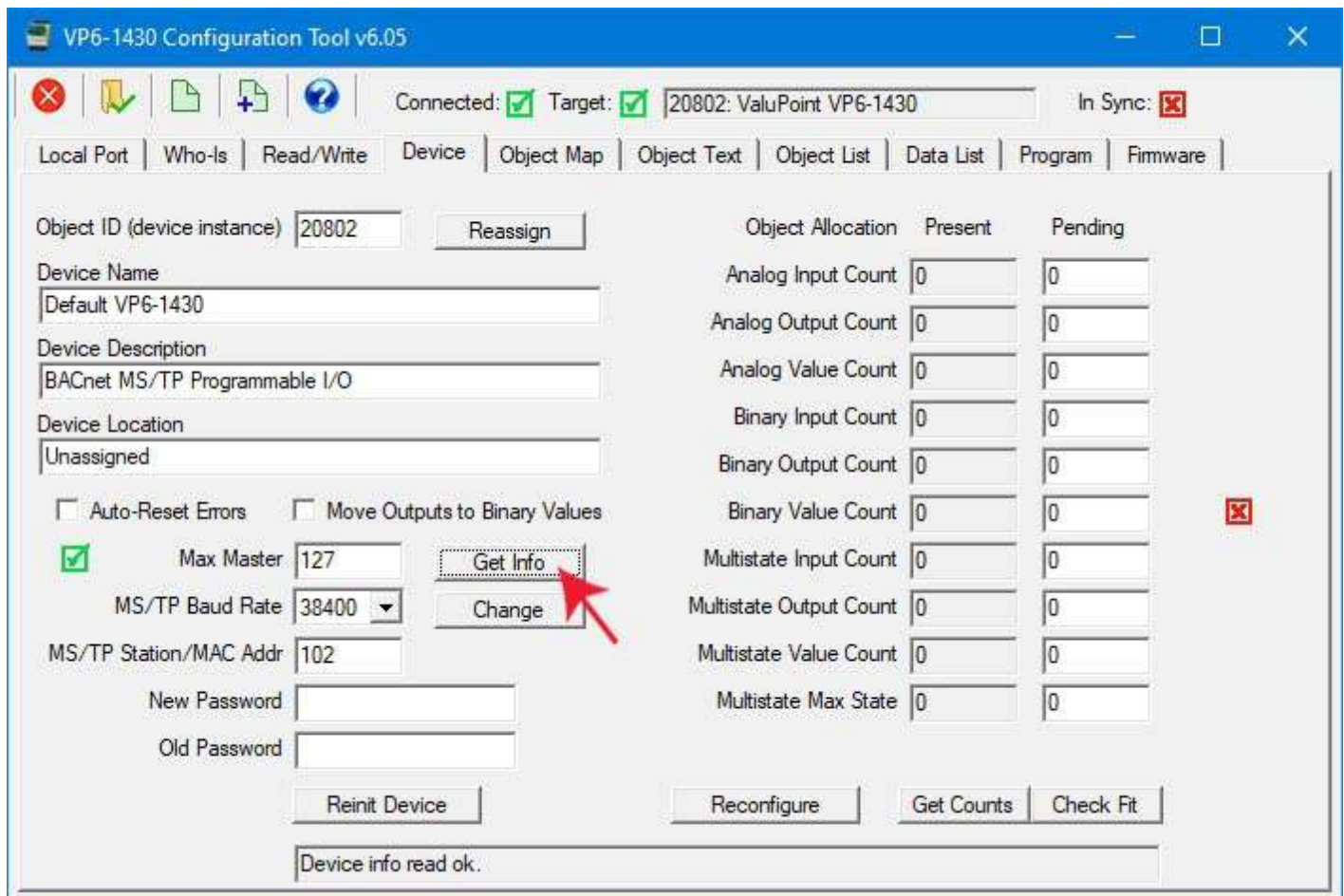


Repeat the device selection just as you did originally except now with the updated device instance.



You are now reconnected with your original device at its now device number. Click the Get Info button to re-read the device object properties just to verify that you are connected.





## 6.4 Get Object Counts from Device

Click 'Get Counts' to read the present object allocation configured in the device. The counts will show up in the 'Present' column. This set of counts tells you how many of each type of object are currently defined in the device. If a BACnet client performs an "auto discovery", this determines how many of each type of object will be discovered.

If you are working offline, i.e. creating a configuration ahead of time without any device connected, then enter the desired object counts in the Pending column, and click Get Counts.

If you will be using Multistate objects, the maximum state value is also set at the bottom of the object count list.

Object ID (device instance) 20801 Reassign

Device Name  
Default VP6-1430

Device Description  
BACnet MS/TP Programmable I/O

Device Location  
Unassigned

Auto-Reset Errors  Move Outputs to Binary Values

Max Master 127 Get Info

MS/TP Baud Rate 38400 Change

MS/TP Station/MAC Addr 102

New Password

Old Password

Reint Device Reconfigure Get Counts Check Fit

Object Allocation	Present	Pending
Analog Input Count	30	30
Analog Output Count	10	10
Analog Value Count	15	15
Binary Input Count	30	30
Binary Output Count	10	10
Binary Value Count	10	10
Multistate Input Count	5	5
Multistate Output Count	5	5
Multistate Value Count	10	10
Multistate Max State	10	10

Device count read ok.

## 6.5 Reconfigure Object Counts

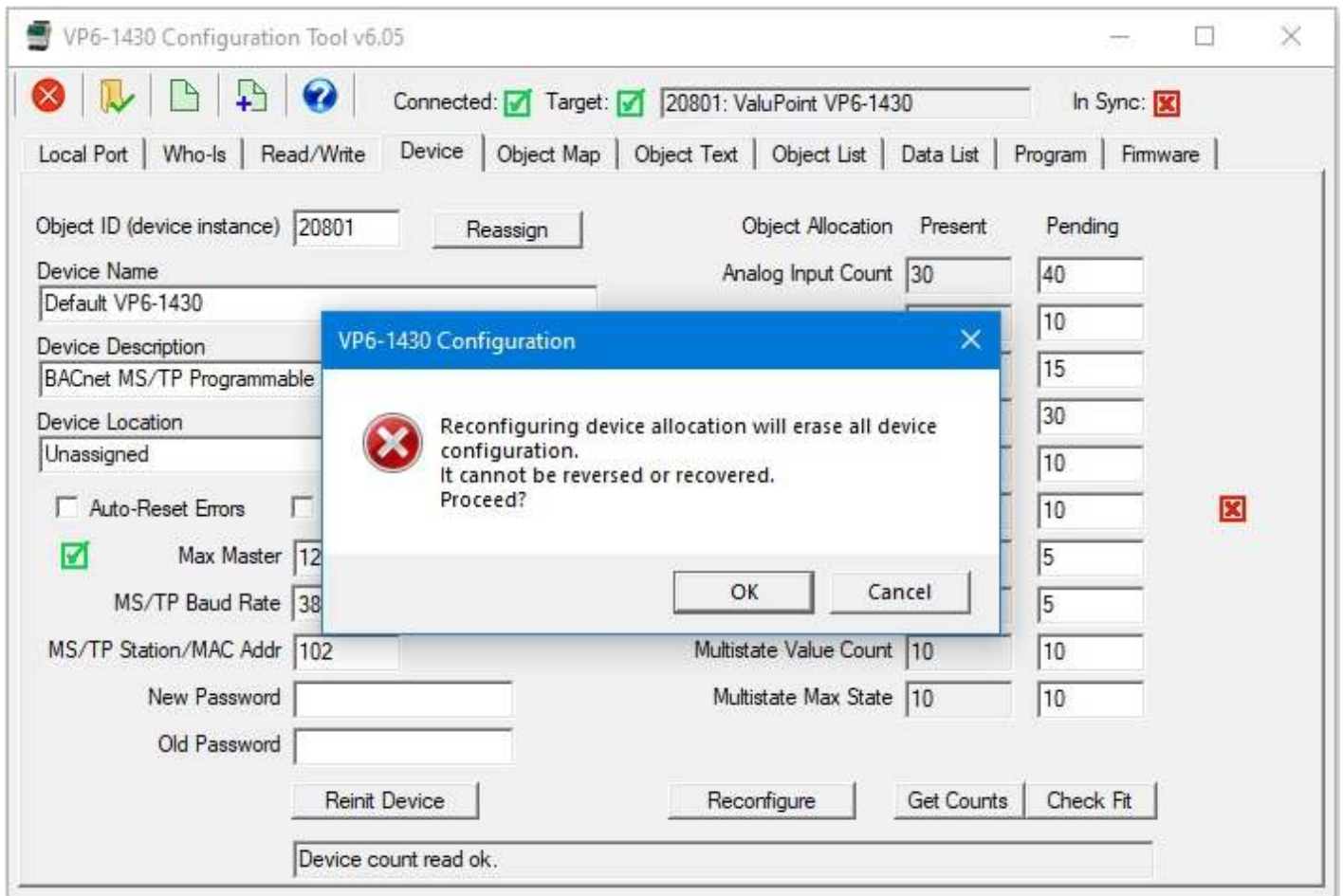
Enter new object counts in the windows in the 'Pending' column.

Object Allocation	Present	Pending
Analog Input Count	30	40
Analog Output Count	10	10
Analog Value Count	15	15
Binary Input Count	30	30
Binary Output Count	10	10
Binary Value Count	10	10
Multistate Input Count	5	5
Multistate Output Count	5	5
Multistate Value Count	10	10
Multistate Max State	10	10

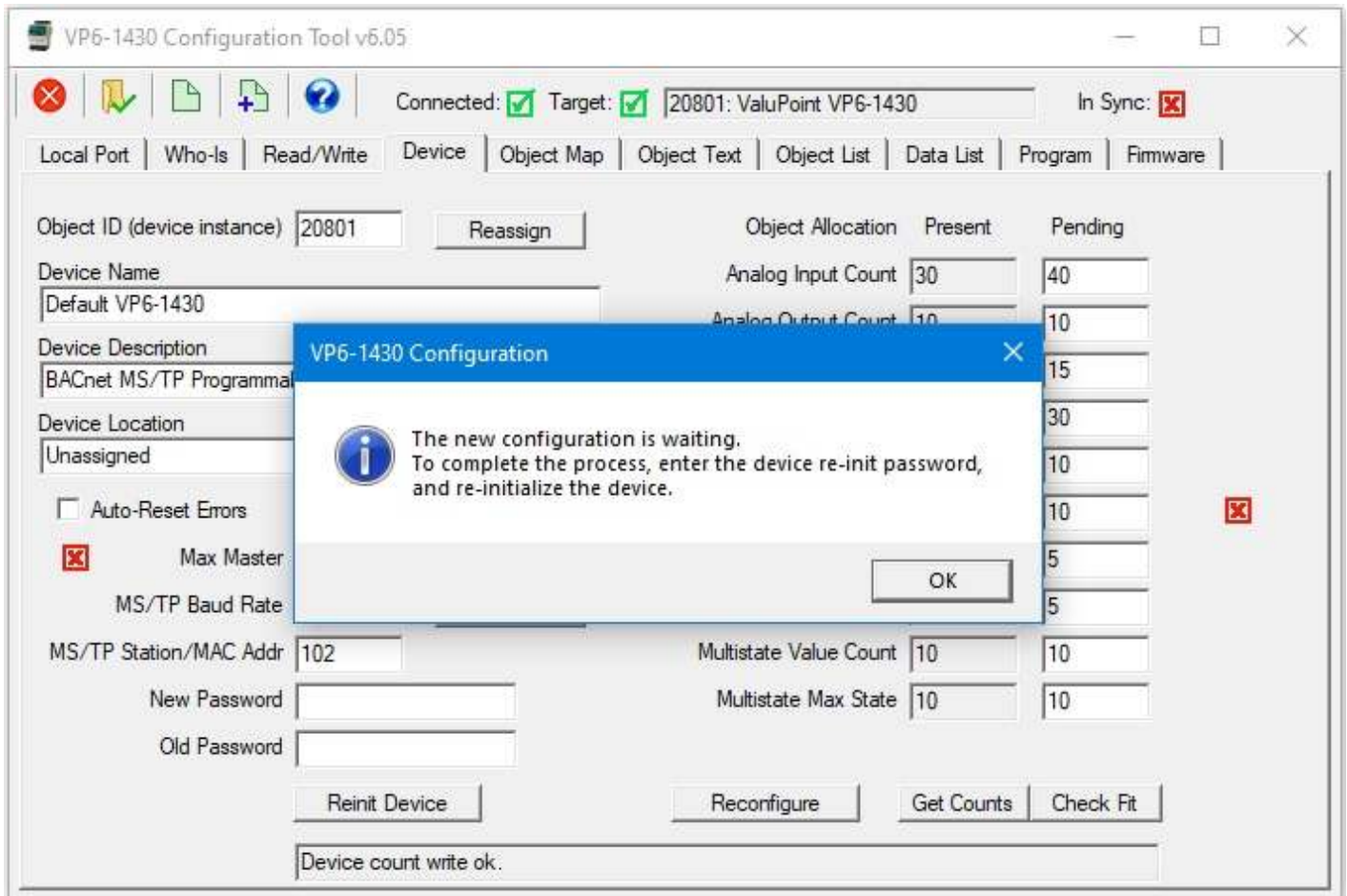
Device count read ok.

Click 'Check Fit' to check to see if the counts are valid.

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.*



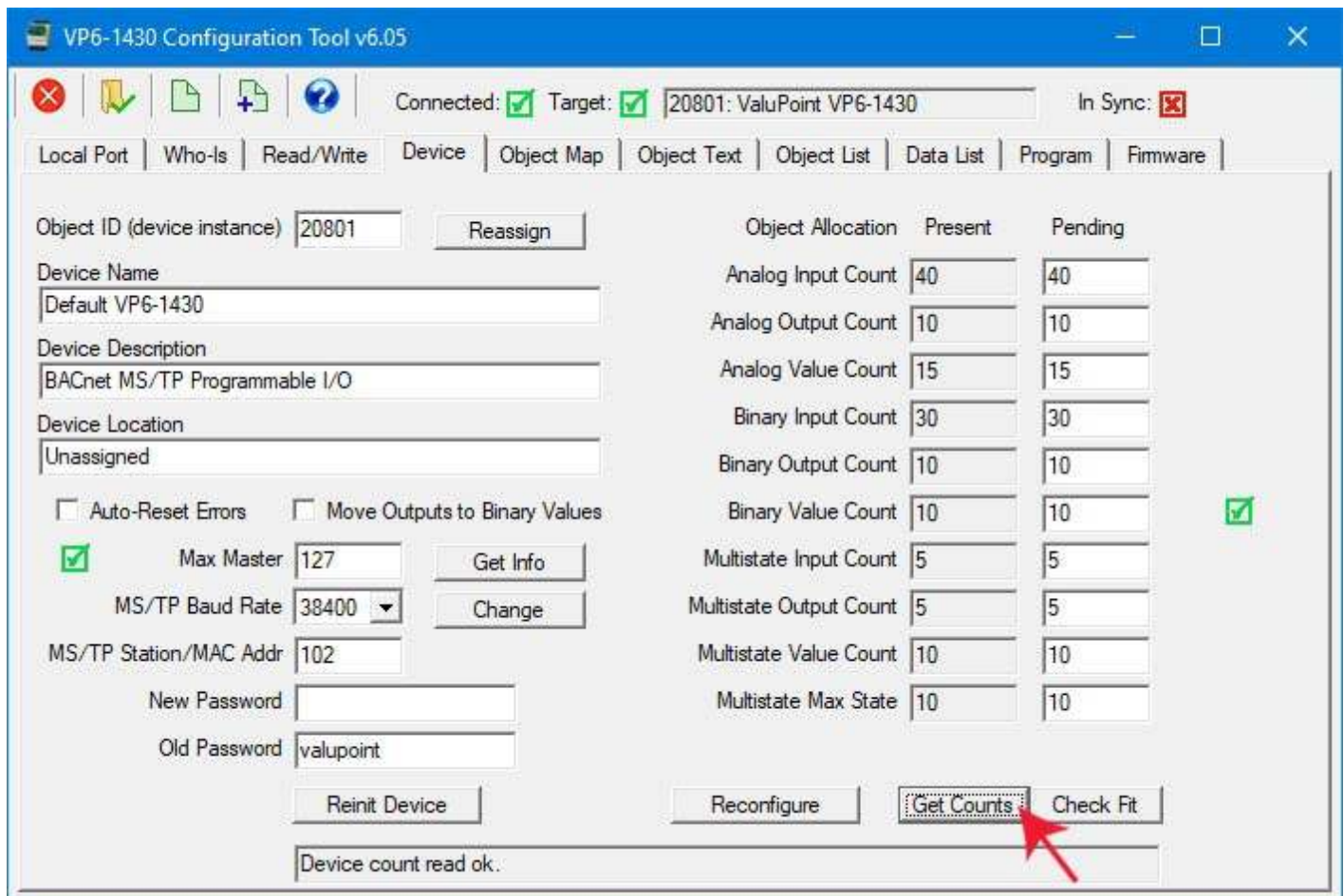
The configuration is now ready to be applied.



At this point, the desired new object counts have been sent to the device, but the device has not yet implemented the change. You need to reinitialize the device to finalize the process. Enter the password (default is "valupoint") and click the Reinit Device button.

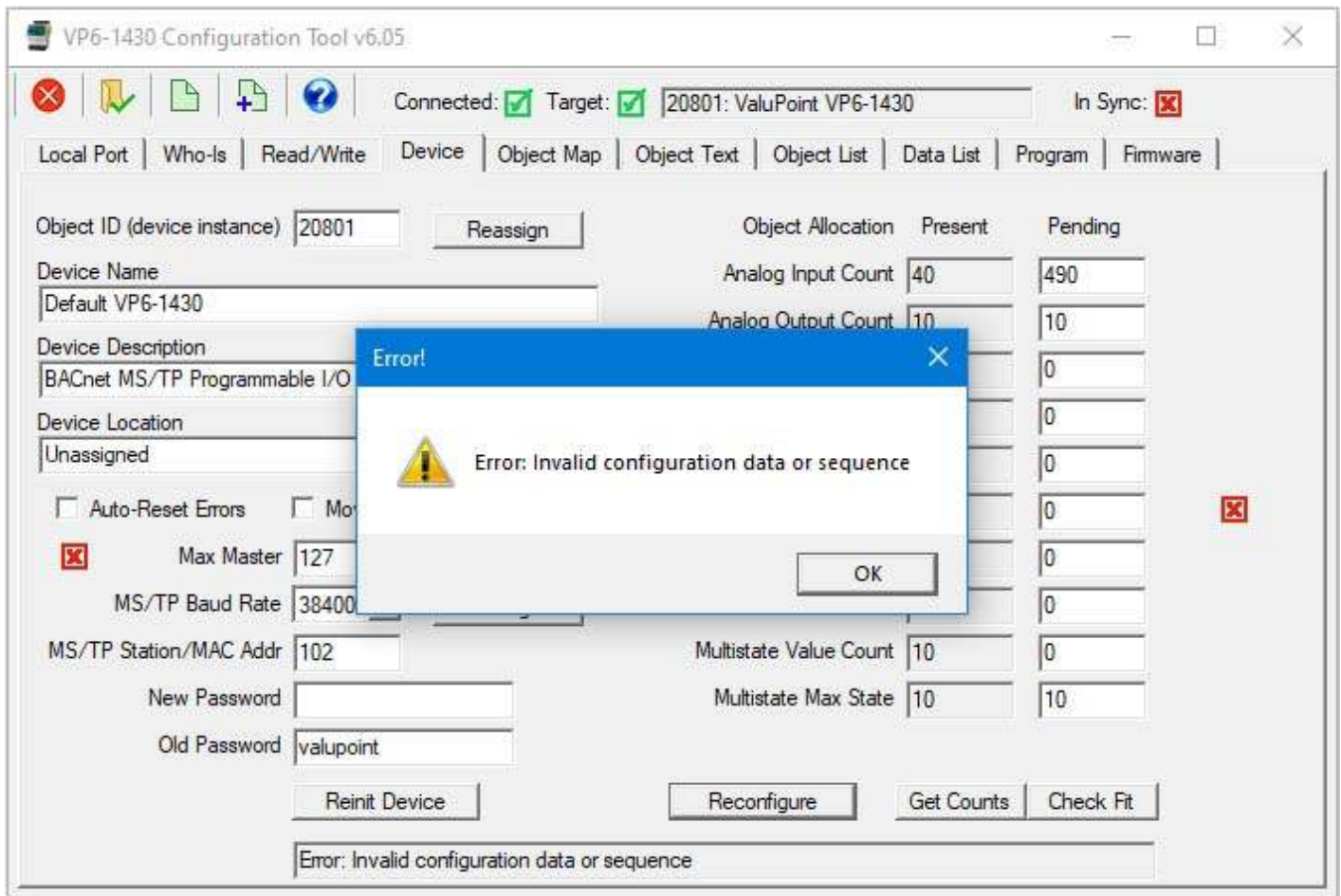
Object Allocation	Present	Pending
Analog Input Count	30	40
Analog Output Count	10	10
Analog Value Count	15	15
Binary Input Count	30	30
Binary Output Count	10	10
Binary Value Count	10	10
Multistate Input Count	5	5
Multistate Output Count	5	5
Multistate Value Count	10	10
Multistate Max State	10	10

You can now click the Get Counts button again to verify the counts in the device. The numbers on the screen will be replaced with whatever is actually in the device, and if all went well, they will be what you thought you had asked for. But if you had failed to enter the correct password, your request was not executed (and there would have been an error message in the status window at the bottom of the screen).



## 6.6 Fixing "Invalid Configuration"

Depending on what the previous object counts had been and what you are trying to change them to, you may get the error indicated in the log window at the bottom of the screen illustrated below. This message appears after clicking OK on the dialog box that appears when you click the Reconfigure button.



If you add up the object counts in the Pending column, the total does not exceed the maximum of 500. So why is this error happening?

The writing of object counts from the configuration tool to the device is largely transparent - you just click the button. But what is really happening is that each count is a property value in the BACnet Device object in the device. To assure that any arbitrary object count can be written, it is necessary to re-process the total allocation as each new count value is written. When the first count of 490 in above example is written to request 490 Analog Input objects, the device is still retaining varying numbers of other objects (see prior examples). That adds up to more than 500, which is over the limit, and the result is the "invalid configuration" error reply.

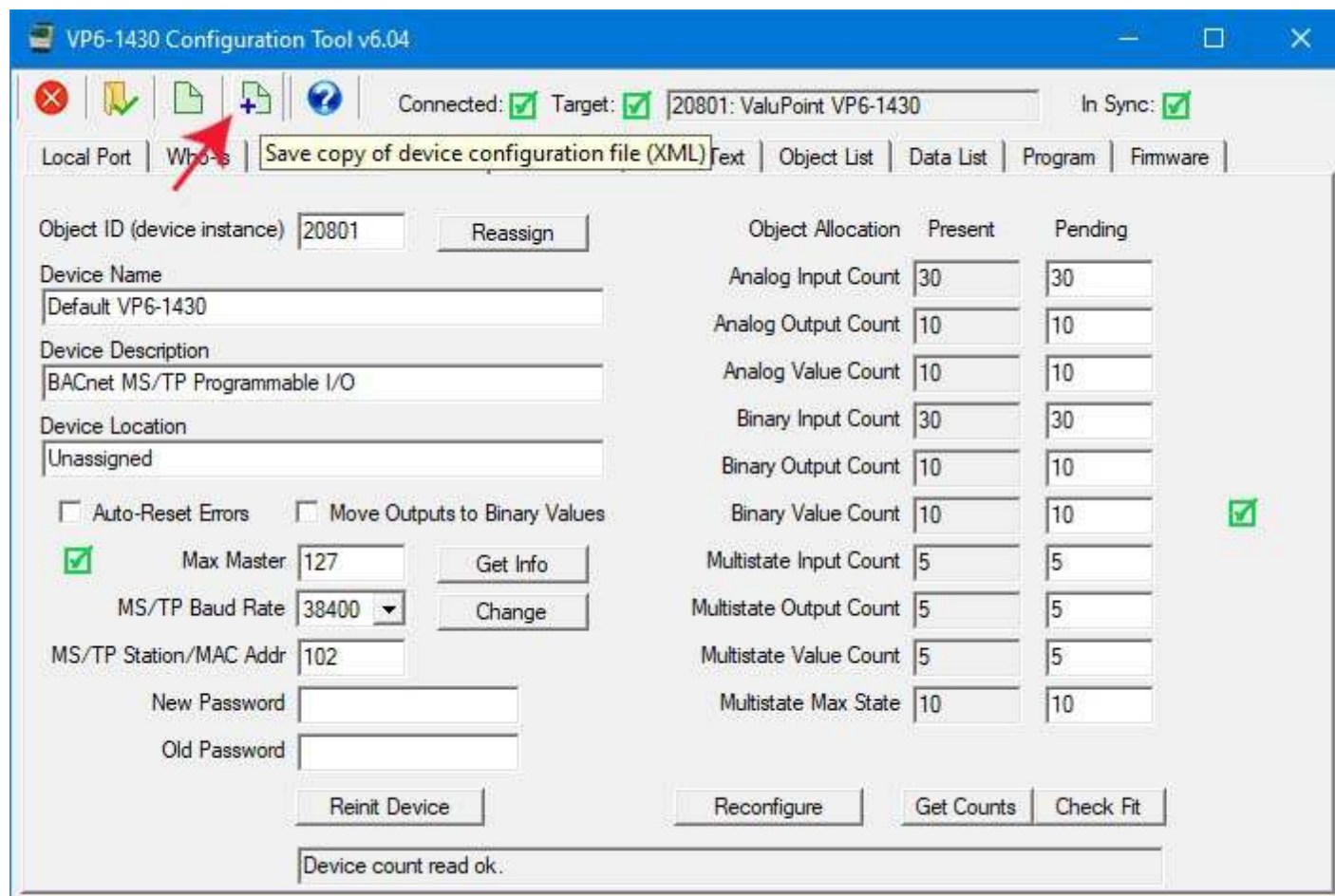
In general, if counts higher on the list are trying to take away from counts lower on the list, you are likely to run into this problem. The solution is to temporarily assign zero objects further down the list. Repeat the object allocation with zero counts as needed. Then go back and retry the counts that were previously not working.

## 6.7 Export and Import Configuration with XML File

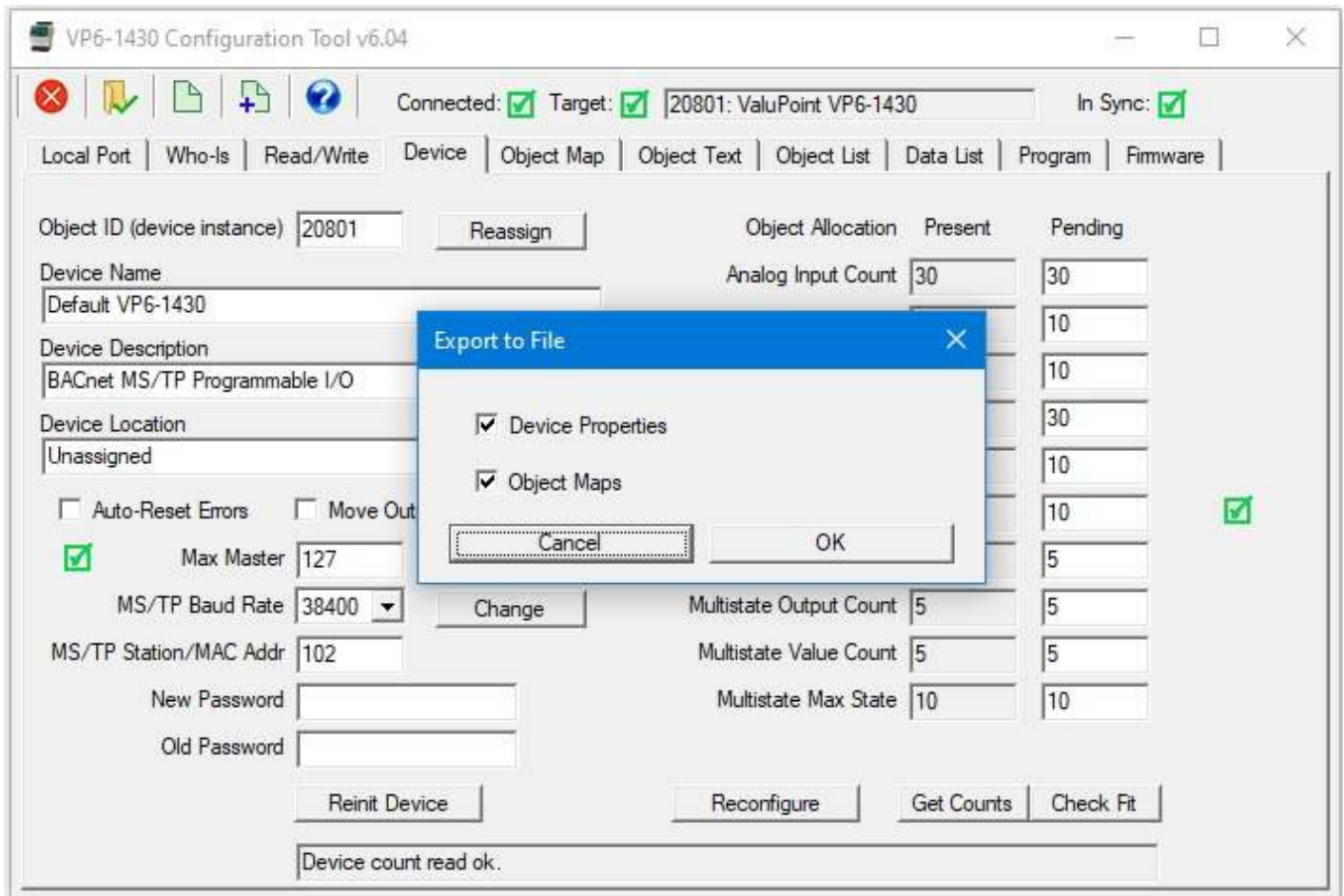
Click the New File icon to save your configuration to a new XML file. You have the option of saving only the device properties, or only the object maps, but the default (and most often preferred) option is to save both.



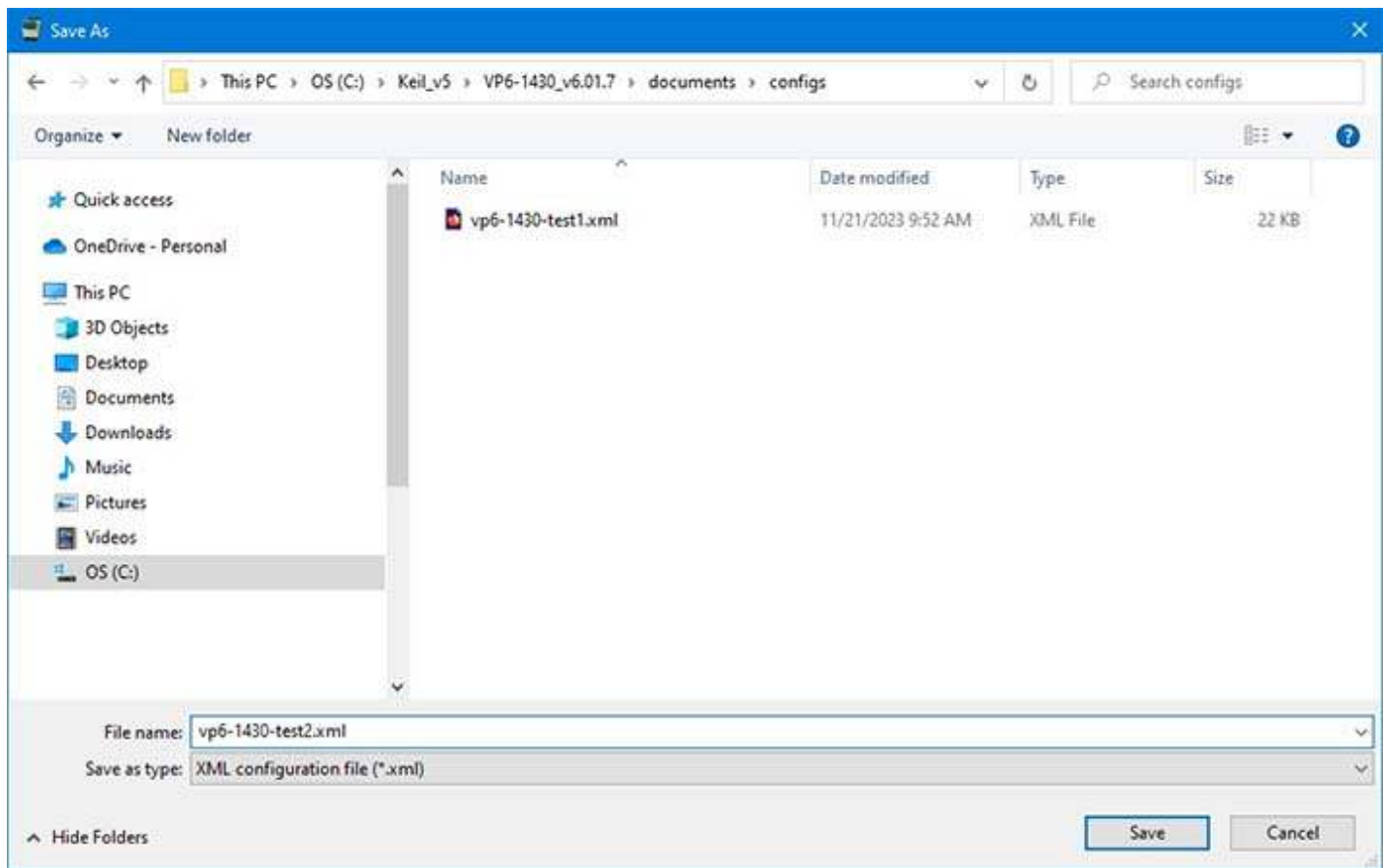
If you want to re-save a previously opened configuration file, following making some changes to configuration, click the first File icon rather than the second icon. The first green icon is "Save" and the second icon is "Save As", where "save as" means create a new file.



You have the option of saving just device properties, just object maps, or both. To copy object mapping to another device, you will usually want to save just the object maps. To create a full backup of the existing configuration for this device, save both.



When you click OK in the "Export to File" dialog, you will next be directed to the Windows file dialog where you may enter a name for your new file.



Once the file is created (or opened), the file name will appear in the top bar. This file will be rewritten if you click the Save icon.

VP6-1430 Configuration Tool v6.04 [vp6-1430-test2.xml]

Connected:  Target:  20801: ValuPoint VP6-1430 In Sync:

Local Port | Who-Is | Read/Write | Device | Object Map | Object Text | Object List | Data List | Program | Firmware

Object ID (device instance) 20801

Device Name  
Default VP6-1430

Device Description  
BACnet MS/TP Programmable I/O

Device Location  
Unassigned

Auto-Reset Errors  Move Outputs to Binary Values

Max Master 127

MS/TP Baud Rate 38400

MS/TP Station/MAC Addr 102

New Password

Old Password

Object Allocation	Present	Pending
Analog Input Count	30	30
Analog Output Count	10	10
Analog Value Count	10	10
Binary Input Count	30	30
Binary Output Count	10	10
Binary Value Count	10	10
Multistate Input Count	5	5
Multistate Output Count	5	5
Multistate Value Count	5	5
Multistate Max State	10	10

File saved.

To load a previously saved configuration into the tool, click the File Open icon.

VP6-1430 Configuration Tool v6.04

Connected:  Target:  20801: ValuPoint VP6-1430 In Sync:

Local P | Open device configuration file (XML) | Object Map | Object Text | Object List | Data List | Program | Firmware

Object ID (device instance) 20801 Reassign

Device Name

Device Description

Device Location

Auto-Reset Errors  Move Outputs to Binary Values

Max Master: 0 Get Info

MS/TP Baud Rate: 38400 Change

MS/TP Station/MAC Addr: 0

New Password

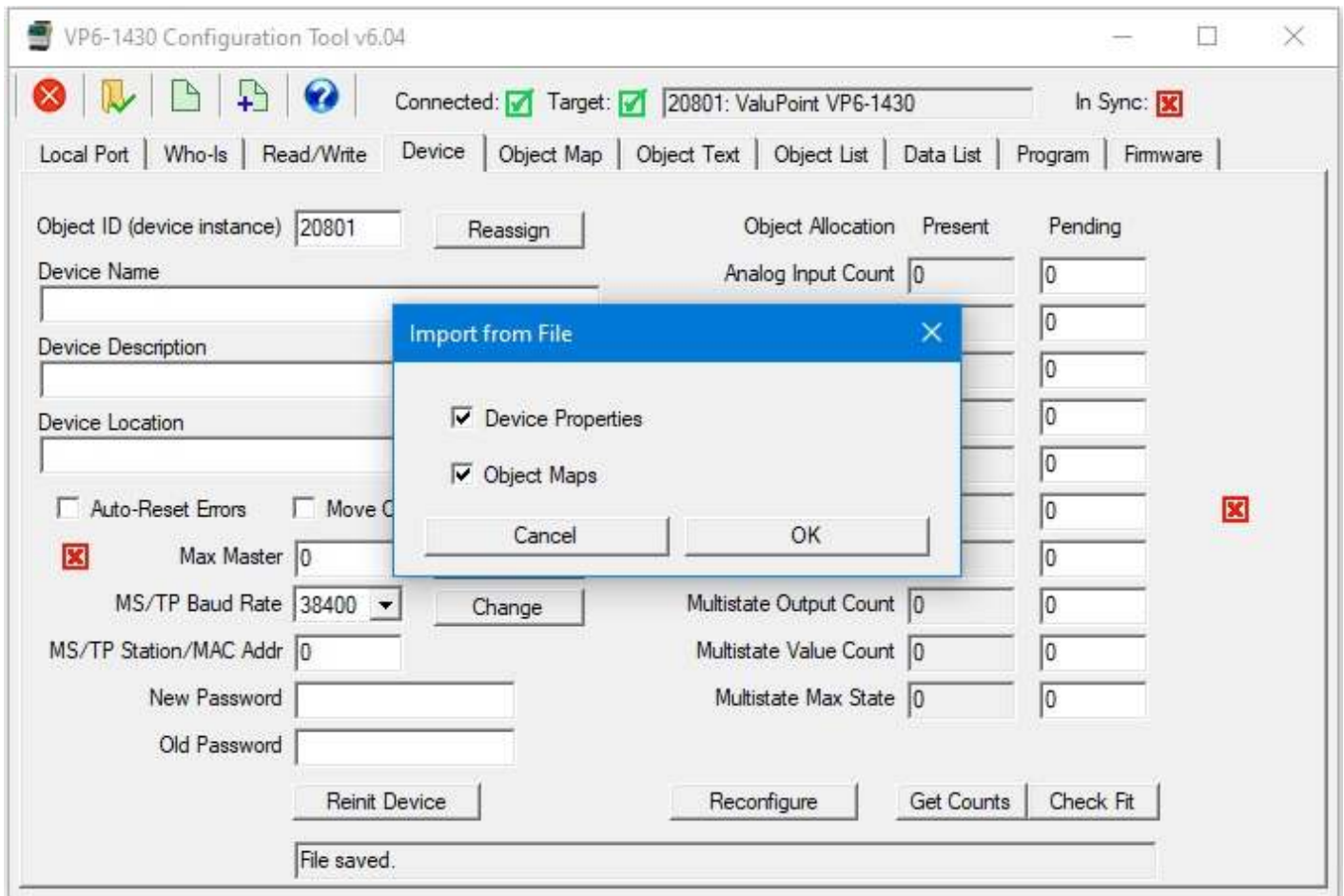
Old Password

Reinit Device Reconfigure Get Counts Check Fit

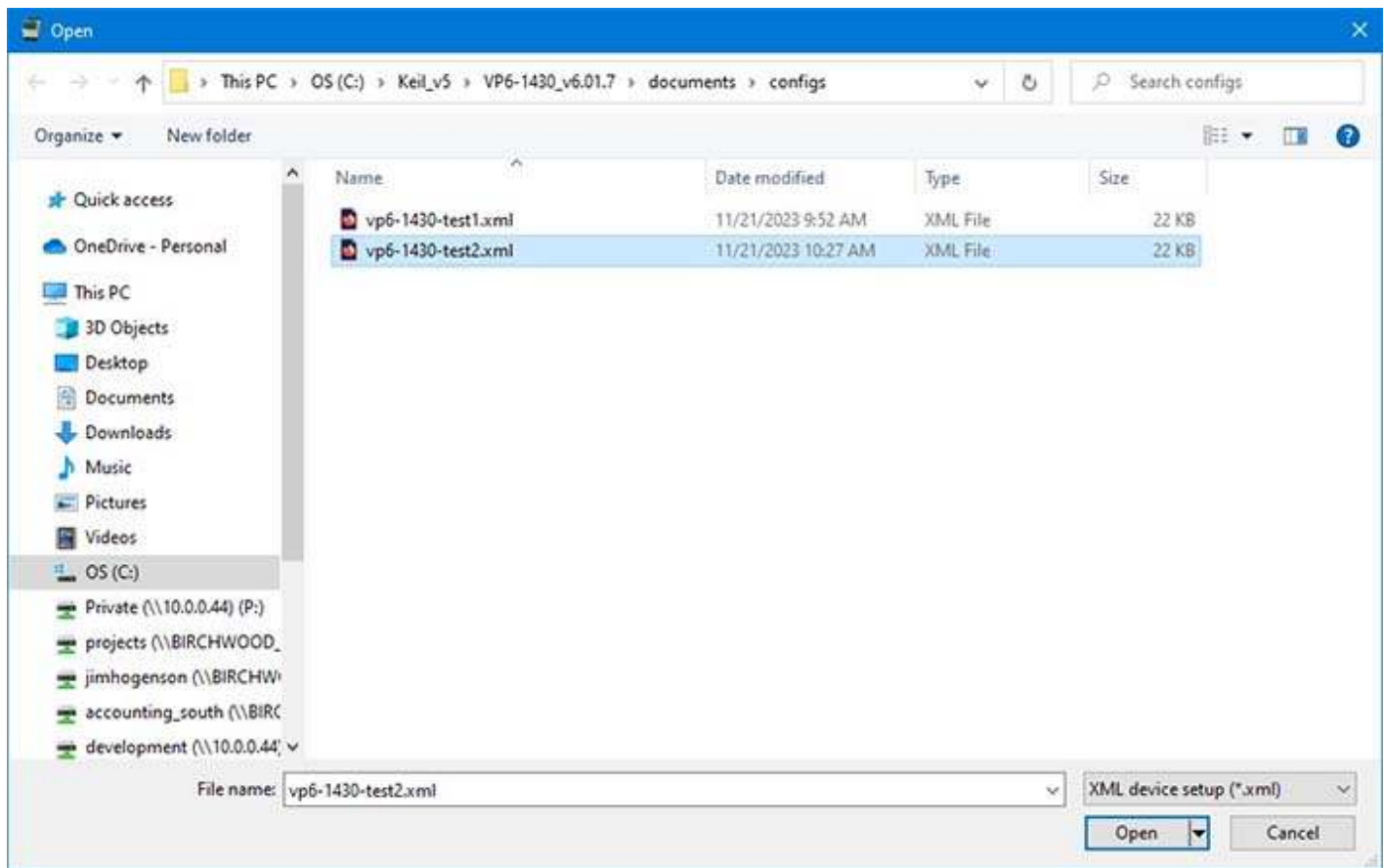
Object Allocation	Present	Pending
Analog Input Count	0	0
Analog Output Count	0	0
Analog Value Count	0	0
Binary Input Count	0	0
Binary Output Count	0	0
Binary Value Count	0	0
Multistate Input Count	0	0
Multistate Output Count	0	0
Multistate Value Count	0	0
Multistate Max State	0	0

File saved.

You will have the option of importing device properties, object maps, or the default of both.



Upon clicking OK in the "Import from File" dialog, you will be directed to the familiar Windows file dialog where you can select the file to be opened.



Select the file and click Open. The various properties will now be populated and the name of the opened file will appear in the top bar.

VP6-1430 Configuration Tool v6.04 [vp6-1430-test2.xml]

Connected:  Target:  20801: ValuPoint VP6-1430 In Sync:

Local Port | Who-Is | Read/Write | Device | Object Map | Object Text | Object List | Data List | Program | Firmware

Object ID (device instance)

Device Name

Device Description

Device Location

Auto-Reset Errors  Move Outputs to Binary Values

Max Master

MS/TP Baud Rate

MS/TP Station/MAC Addr

New Password

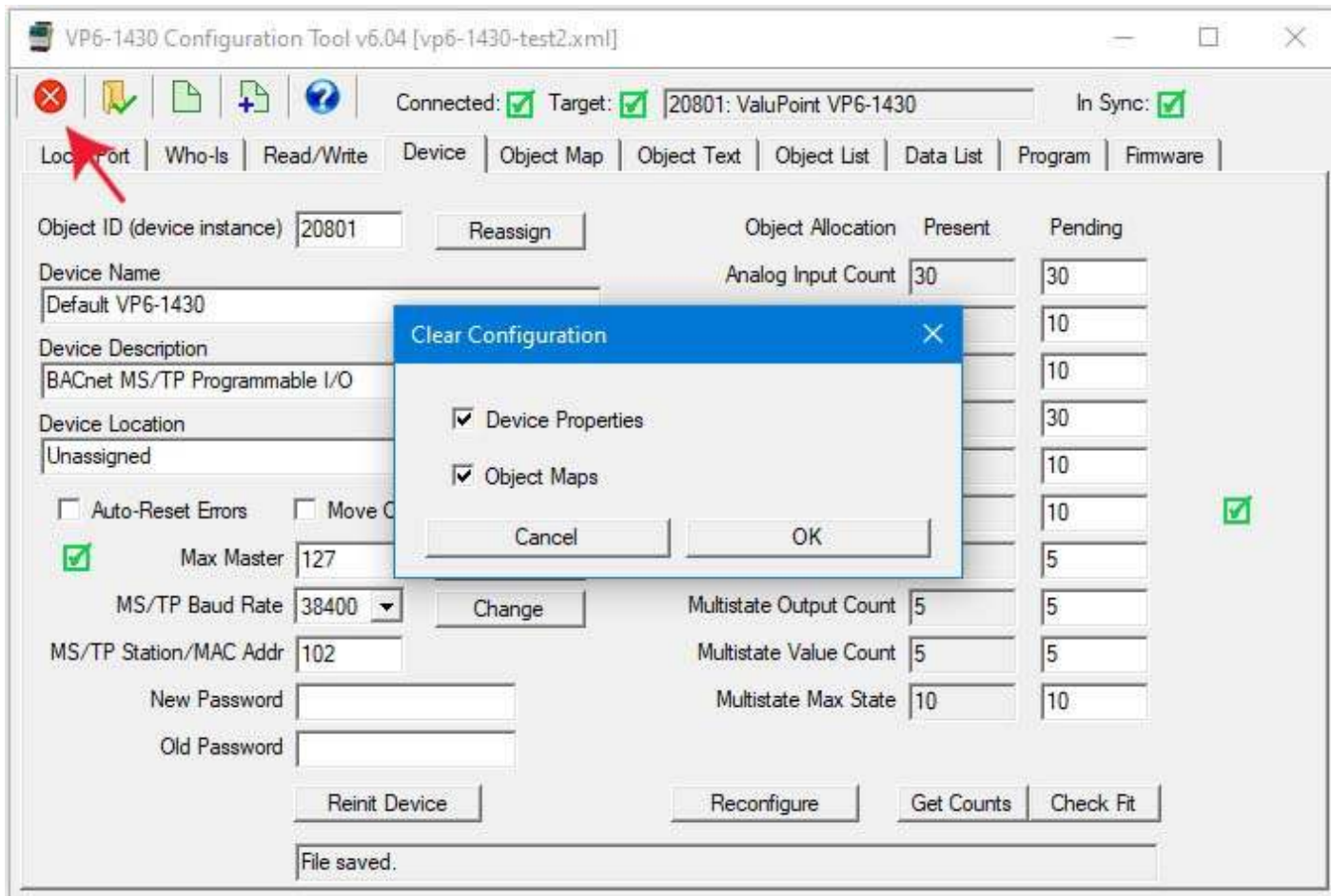
Old Password

Object Allocation	Present	Pending
Analog Input Count	<input type="text" value="30"/>	<input type="text" value="30"/>
Analog Output Count	<input type="text" value="10"/>	<input type="text" value="10"/>
Analog Value Count	<input type="text" value="10"/>	<input type="text" value="10"/>
Binary Input Count	<input type="text" value="30"/>	<input type="text" value="30"/>
Binary Output Count	<input type="text" value="10"/>	<input type="text" value="10"/>
Binary Value Count	<input type="text" value="10"/>	<input type="text" value="10"/>
Multistate Input Count	<input type="text" value="5"/>	<input type="text" value="5"/>
Multistate Output Count	<input type="text" value="5"/>	<input type="text" value="5"/>
Multistate Value Count	<input type="text" value="5"/>	<input type="text" value="5"/>
Multistate Max State	<input type="text" value="10"/>	<input type="text" value="10"/>

File loaded.

To wipe the slate completely clean in the configuration tool, click the Clear icon, which is the first icon on the toolbar. You will be given the option of clearing device properties, object maps, or both. Or just click Cancel if you clicked the Clear icon by mistake.





Clear everything will clear all but the device instance.



## 7 Object Map Page

When connecting to a device for the first time, it will be quickest 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 (a few hundred times for a fully loaded device). Better yet, just click 'Read All Config' on the Who-Is page after selecting your target device.

This page is where you enter various parameters for each object, basically telling the system what each object is supposed to do. You will only be able to configure as many objects as you have declared on the Device page, so be sure you check that page first.

Objects will typically do one of two things: (1) Be a data point associated with one of the physical I/O points; (2) Be a client that reads data from another BACnet device and/or writes data to another BACnet device. There is a third less commonly used option, "global object", which basically does nothing but wait for something else to write data to it.

### 7.1 Common Parameters

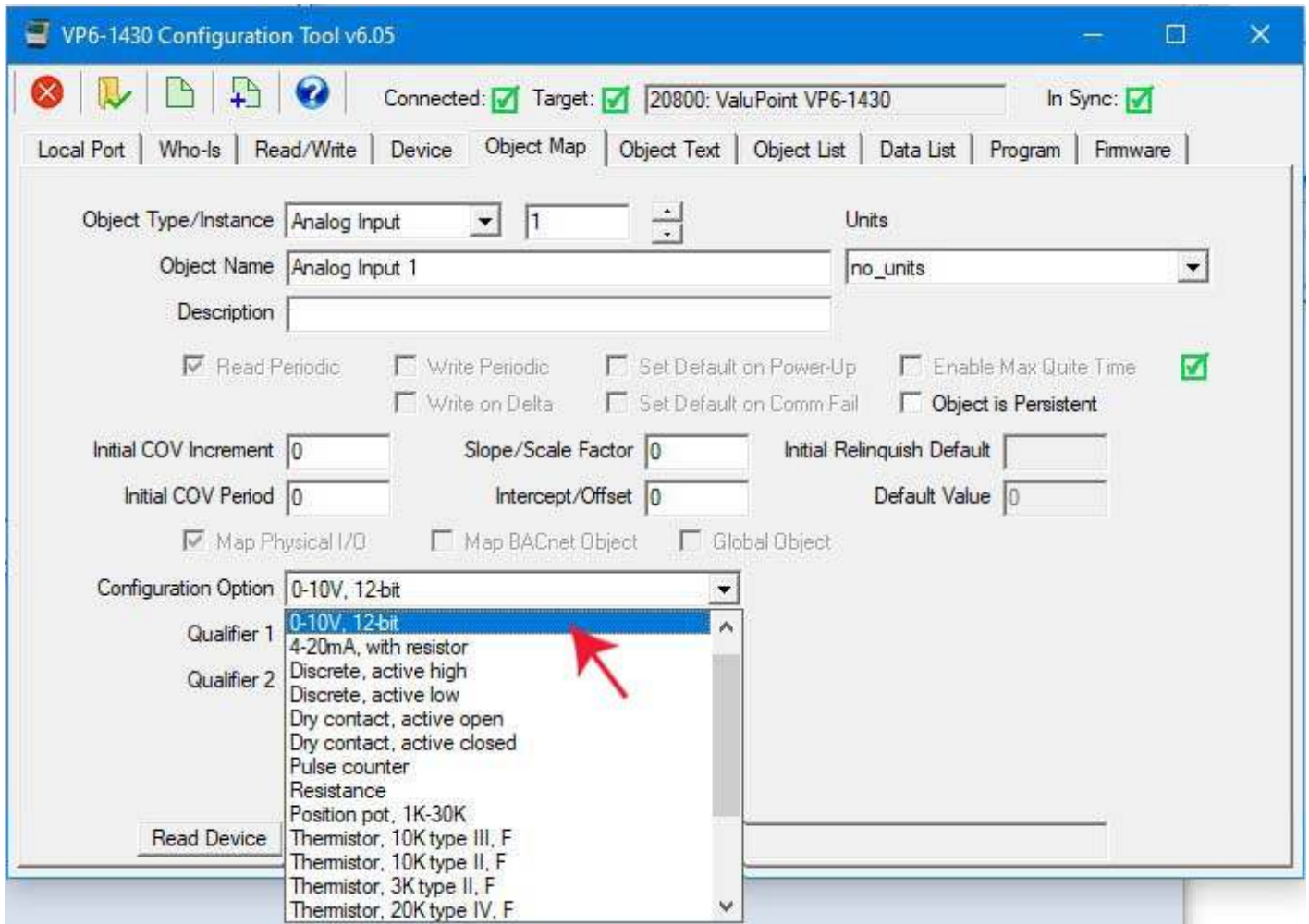
The following common parameters will appear in the top portion of the screen for all types of object uses.

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 and does not have to be unique.
Units	Select a BACnet units type from the list.
Read Periodic	Check this box if this BACnet object will be periodically reading from another BACnet device. 'Read' is valid for Input and Value type objects.
Write Periodic	Check this box if this BACnet object will be periodically writing to another BACnet device. 'Write' is valid for Output and Value type objects.
Write on Delta	Check this box if this BACnet object will be writing to another device only when this 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 this box if this object should assume the default value every time the ValuPoint powers up.
Set Default on Comm Fail	<p>Check this box if the object should assume the default value when communication with the other device has failed some number of times (when Read or Write are checked).</p> <p>NOTE: If 'Set Default on Comm Fail' is enabled after fault already exists, the fault must clear and repeat before the default will be applied. The default value is applied only one time upon transition to fault state to avoid repeated COV notifications.</p>
Enable Max Quiet Time	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 remote device will be re-written anyway.
Object is Persistent	Checking this box means the data value of the object will be retained through restart or power cycle, rather than reset to zero. This is applicable to only certain objects. Objects mapped to physical inputs, for certain configurations, will not honor the Persistent flag. For example, an Analog Input will always reflect the actual input value when configured for 0-10V input. However, the persistent option does apply when the input is configured for pulse counting. When the count is persistent, the count will be retained through restart or power cycle; however, to extend the life of the non-volatile memory, the persistent value is only saved once every 15 minutes.
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 COV Period	The initial COV Period will take effect when the ValuPoint is first powered up. After that, any BACnet client can write a new COV Period to the object. The COV Period produces a periodic COV notification regardless of value change. COV Period is valid for all object types. Enter zero to disable periodic COV and provide notification only upon change specified by the increment.

Slope/ Scale Factor	Scaling applies the formula $y=mx+b$ . When reading from a physical input or a remote BACnet device, 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 remote BACnet device, 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 remote 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.
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 (Output) object, and the Present Value of the object will assume this value when all 16 priorities have been relinquished.
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.

## 7.2 Physical Hardware I/O Parameters



The following configuration properties apply to physical I/O points and will be available when Map Physical I/O is checked.

<p>Map Physical I/O</p>	<p>This box will be automatically checked when applicable. Physical I/O points are automatically and permanently assigned as follows: Physical A/UI inputs 1 through 12 are assigned to Analog Input objects 1 through 12, and are mirrored as state inputs in Binary Input objects 1 through 12. The relay outputs DO 1 and 2 are assigned by default to Binary Output objects 1 and 2, but can be moved to Binary Value objects 1 and 2 if selected on the Device page.</p>
<p>Configuration Option</p>	<p>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:</p> <ul style="list-style-type: none"> <li>0-10V, 12-bit resolution</li> <li>4-20mA, with resistor</li> <li>Discrete, active high</li> <li>Discrete, active low</li> <li>Dry contact, active open</li> <li>Dry contact, active closed</li> </ul>

Pulse counter  
Resistance  
Position pot, 1K-30K

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

Dedicated hardware is available for pulse counting on channels 5, 6, 7, and 8. The only limiting factor on maximum pulse rate on these inputs is the noise filtering on the inputs. The inputs have been verified to count at up to 1kHz provided the signal amplitude is sufficient. Pulse counting is supported on the remaining input channels, but the counting is done by software and therefore the rate is limited to about 2Hz.

Qualifier 1	<p>Enter the configuration qualifier value, if applicable, for the selected configuration. Qualifiers are required only in the following modes:</p> <p><i>4-20mA modes:</i> 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.</p> <p><i>Discrete and Dry Contact modes:</i> The qualifier is a threshold between 1% and 99% at which the input should trip from off to on or vice versa. The A/UI inputs are specified as 0-10V inputs. Therefore, since discrete inputs are sampled as analog values and compared to a threshold, the qualifier here is a percentage of 10V for the trip point. A value of 50% will mean a threshold of around 5V.</p> <p><i>Position Pot:</i> 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.</p>
Qualifier 2	Enter a second configuration qualifier value if instructed to do so. This qualifier is reserved for future use.

The screen shot below shows a typical configuration for reading a 0-20mA current loop value. With no slope or intercept values entered, the Present Value will be the milliamp reading observed.

Analog input scaling follows the standard slope intercept formula  $y=mx+b$ . To solve for slope  $m$  you would use the formula  $m = (y_2-y_1)/(x_2-x_1)$ . Once you have slope  $m$ , solve for intercept  $b$  using  $b = y-mx$ .

Now, for example, let's suppose we want the numeric range for our input to be 0 to 100 (%), corresponding to 4 to 20mA. In other words, 4mA should produce 0, and 20mA should produce 100. The  $x$  values are 4 and 20, and the  $y$  values are 0 and 100.

$$m = (100 - 0) / (20 - 4) = 6.25$$

$$b = 100 - (6.25 * 20) = -25$$

With these slope and intercept numbers, 4mA should give a Present Value of 0, 12mA should read 50%, and 20mA should read 100%.



The screenshot displays the 'Object Map' configuration window for a ValuPoint VP6-1430 device. The window title is 'VP6-1430 Configuration Tool v6.05'. The status bar indicates 'Connected: [checked]', 'Target: [checked]', and 'In Sync: [unchecked]'. The target address is '20800: ValuPoint VP6-1430'. The 'Object Map' tab is selected, showing the configuration for 'Analog Input 1'.

Configuration details for 'Analog Input 1':

- Object Type/Instance: Analog Input, 1
- Units: no\_units
- Object Name: Analog Input 1
- Description: (empty)
- Read Periodic:  (checked)
- Write Periodic:  (unchecked)
- Set Default on Power-Up:  (unchecked)
- Enable Max Quiet Time:  (checked)
- Write on Delta:  (unchecked)
- Set Default on Comm Fail:  (unchecked)
- Object is Persistent:  (unchecked)
- Initial COV Increment: 0
- Slope/Scale Factor: 0
- Initial Relinquish Default: (empty)
- Initial COV Period: 0
- Intercept/Offset: 0
- Default Value: 0
- Map Physical I/O:  (checked)
- Map BACnet Object:  (unchecked)
- Global Object:  (unchecked)
- Configuration Option: 4-20mA, with resistor
- Qualifier 1: 499
- Qualifier 2: 0

Buttons at the bottom: Read Device, Write Device.

The screen shot below shows configuration for high speed pulse counting with count set to persistent, meaning this is effectively a totalizer.

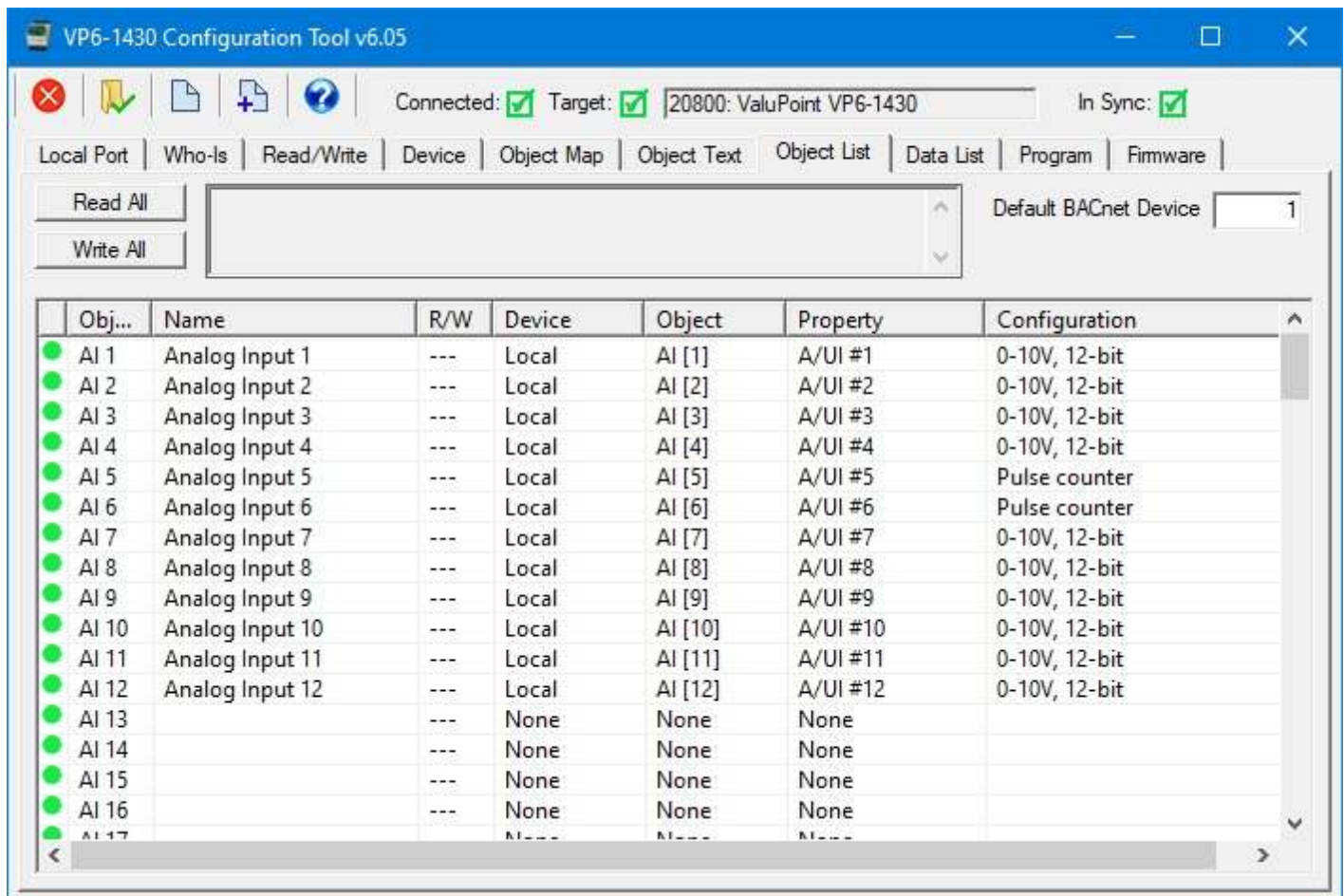
The screenshot displays the 'Object List' page in the 'VP6-1430 Configuration Tool v6.05'. The interface includes a top navigation bar with tabs for 'Local Port', 'Who-Is', 'Read/Write', 'Device', 'Object Map', 'Object Text', 'Object List', 'Data List', 'Program', and 'Firmware'. The 'Object List' tab is active, showing configuration details for an object.

Key configuration fields and options include:

- Object Type/Instance:** Analog Input (dropdown), 5 (spin box), Units: no\_units (dropdown)
- Object Name:** Analog Input 5 (text box)
- Description:** (empty text box)
- Options:**
  - Read Periodic
  - Write Periodic
  - Set Default on Power-Up
  - Enable Max Quiet Time
  - Write on Delta
  - Set Default on Comm Fail
  - Object is Persistent
- Parameters:**
  - Initial COV Increment: 0 (spin box)
  - Slope/Scale Factor: 0 (spin box)
  - Initial Relinquish Default: (empty spin box)
  - Initial COV Period: 0 (spin box)
  - Intercept/Offset: 0 (spin box)
  - Default Value: 0 (spin box)
- Mapping Options:**
  - Map Physical I/O
  - Map BACnet Object
  - Global Object
- Configuration Option:** Pulse counter (dropdown)
- Qualifiers:** Qualifier 1: 0 (spin box), Qualifier 2: 0 (spin box)

At the bottom, there are 'Read Device' and 'Write Device' buttons, and a status bar showing 'OBJECT\_ANALOG\_INPUT #1 -- read ok.'

An example of the Object List page with two channels set for pulse counting.



VP6-1430 Configuration Tool v6.05

Connected:  Target:  20800: ValuPoint VP6-1430 In Sync:

Local Port | Who-Is | Read/Write | Device | Object Map | Object Text | Object List | Data List | Program | Firmware

Read All | Write All | Default BACnet Device: 1

Obj...	Name	R/W	Device	Object	Property	Configuration
AI 1	Analog Input 1	---	Local	AI [1]	A/UI #1	0-10V, 12-bit
AI 2	Analog Input 2	---	Local	AI [2]	A/UI #2	0-10V, 12-bit
AI 3	Analog Input 3	---	Local	AI [3]	A/UI #3	0-10V, 12-bit
AI 4	Analog Input 4	---	Local	AI [4]	A/UI #4	0-10V, 12-bit
AI 5	Analog Input 5	---	Local	AI [5]	A/UI #5	Pulse counter
AI 6	Analog Input 6	---	Local	AI [6]	A/UI #6	Pulse counter
AI 7	Analog Input 7	---	Local	AI [7]	A/UI #7	0-10V, 12-bit
AI 8	Analog Input 8	---	Local	AI [8]	A/UI #8	0-10V, 12-bit
AI 9	Analog Input 9	---	Local	AI [9]	A/UI #9	0-10V, 12-bit
AI 10	Analog Input 10	---	Local	AI [10]	A/UI #10	0-10V, 12-bit
AI 11	Analog Input 11	---	Local	AI [11]	A/UI #11	0-10V, 12-bit
AI 12	Analog Input 12	---	Local	AI [12]	A/UI #12	0-10V, 12-bit
AI 13		---	None	None	None	
AI 14		---	None	None	None	
AI 15		---	None	None	None	
AI 16		---	None	None	None	
AI 17		---	None	None	None	

Note that Binary Inputs 1 through 12 are fixed and assigned to echo the state of the physical inputs A/UI 1 through 12. When the A/UI input is configured for a state, e.g. "Dry contact, active closed", the state value of 0 or 1 is only accessible as a floating point value through the Analog Input object. To access this same state as a Binary object, use the Binary Input mirror of the object instead.

The screenshot shows the VP6-1430 Configuration Tool v6.05 interface. The top bar indicates the tool is connected to a ValuPoint VP6-1430 device. The Object Map tab is selected, displaying a table of objects. The table has columns for Object ID, Name, R/W, Device, Object, Property, and Configuration. The BI 1 object is highlighted, showing its configuration as 'Fixed'.

Obj...	Name	R/W	Device	Object	Property	Configuration
AV 9		---	None	None	None	
AV 10		---	None	None	None	
BI 1	No name [BI 1]	---	Local	BI [1]	A/UI #1	Fixed
BI 2	No name [BI 2]	---	Local	BI [2]	A/UI #2	Fixed
BI 3	No name [BI 3]	---	Local	BI [3]	A/UI #3	Fixed
BI 4	No name [BI 4]	---	Local	BI [4]	A/UI #4	Fixed
BI 5	No name [BI 5]	---	Local	BI [5]	A/UI #5	Fixed
BI 6	No name [BI 6]	---	Local	BI [6]	A/UI #6	Fixed
BI 7	No name [BI 7]	---	Local	BI [7]	A/UI #7	Fixed
BI 8	No name [BI 8]	---	Local	BI [8]	A/UI #8	Fixed
BI 9	No name [BI 9]	---	Local	BI [9]	A/UI #9	Fixed
BI 10	No name [BI 10]	---	Local	BI [10]	A/UI #10	Fixed
BI 11	No name [BI 11]	---	Local	BI [11]	A/UI #11	Fixed
BI 12	No name [BI 12]	---	Local	BI [12]	A/UI #12	Fixed
BI 13		---	None	None	None	
BI 14		---	None	None	None	
BI 15		---	None	None	None	

The Default Value can be assigned to any Output or Value type of object. In the example pictured below, relay #1 (BO 1) is always going to turn on as soon as the VP6-1430 powers up.

The screenshot displays the 'Object Map' configuration window for a ValuPoint VP6-1430 device. The window title is 'VP6-1430 Configuration Tool v6.05'. At the top, it shows 'Connected: [checked]', 'Target: [checked] 20801: ValuPoint VP6-1430', and 'In Sync: [checked]'. Below this is a navigation bar with tabs: 'Local Port', 'Who-Is', 'Read/Write', 'Device', 'Object Map' (selected), 'Object Text', 'Object List', 'Data List', 'Program', and 'Firmware'. The main configuration area includes:

- Object Type/Instance:** Binary Output (dropdown), 1 (spin box), Units: no\_units (dropdown)
- Object Name:** No name [BO 1] (text box)
- Description:** (empty text box)
- Behavioral Options:**
  - Read Periodic
  - Write Periodic
  - Set Default on Power-Up
  - Enable Max Quiet Time [checked]
  - Write on Delta
  - Set Default on Comm Fail
  - Object is Persistent
- COV Parameters:**
  - Initial COV Increment: (empty)
  - Slope/Scale Factor: 0 (spin box)
  - Initial Relinquish Default: 0 (spin box)
  - Initial COV Period: 0 (spin box)
  - Intercept/Offset: 0 (spin box)
  - Default Value: 1 (spin box)
- Mapping Options:**
  - Map Physical I/O
  - Map BACnet Object
  - Global Object

At the bottom, there are 'Read Device' and 'Write Device' buttons, followed by a large empty text box.

The alternate method of setting a default value for an Output object is to set the initial relinquish default as illustrated below. The screen below and the screen above accomplish exactly the same thing for a Binary Output object.

The screenshot shows the 'VP6-1430 Configuration Tool v6.05' window. The 'Object Map' tab is selected. The interface includes a toolbar with icons for file operations and a status bar showing 'Connected: [checked]', 'Target: [checked]', and 'In Sync: [checked]'. The target address is '20801: ValuPoint VP6-1430'. Below the toolbar are tabs for 'Local Port', 'Who-Is', 'Read/Write', 'Device', 'Object Map', 'Object Text', 'Object List', 'Data List', 'Program', and 'Firmware'. The main configuration area contains the following fields and options:

- Object Type/Instance: Binary Output (dropdown), 1 (input)
- Object Name: No name [BO 1] (text box)
- Units: no\_units (dropdown)
- Description: (empty text box)
- Read Periodic:
- Write Periodic:
- Set Default on Power-Up:
- Enable Max Quiet Time:
- Write on Delta:
- Set Default on Comm Fail:
- Object is Persistent:
- Initial COV Increment: (empty text box)
- Slope/Scale Factor: 0 (text box)
- Initial Relinquish Default: 1 (text box)
- Initial COV Period: 0 (text box)
- Intercept/Offset: 0 (text box)
- Default Value: 0 (text box)
- Map Physical I/O:
- Map BACnet Object:
- Global Object:

At the bottom, there are 'Read Device' and 'Write Device' buttons and a large empty text box.

### 7.3 BACnet Client Parameters

These configuration parameters are used when you want to have the VP6-1430 interact with other BACnet devices on the MS/TP network.

VP6-1430 Configuration Tool v6.04

Connected:  Target:  20801: ValuPoint VP6-1430 In Sync:

Local Port | Who-Is | Read/Write | Device | Object Map | Object Text | Object List | Data List | Program | Firmware

Object Type/Instance: Analog Value 1 Units: square\_meters

Object Name: Analog Value 1

Description: Test client read

Read Periodic  Write Periodic  Set Default on Power-Up  Enable Max Quiet Time

Write on Delta  Set Default on Comm Fail  Object is Persistent

Initial COV Increment: 0 Slope/Scale Factor: 0 Initial Relinquish Default: 0

Initial COV Period: 0 Intercept/Offset: 0 Default Value: 0

Map Physical I/O  Map BACnet Object  Global Object

Remote Device Instance: 40162 Property: Present Value 85 Bit Num: 0

Object Type: Analog Value Instance: 1 Data Type: Null

Mac address (no who-is): 0 Array Index: -1 Priority: None

Poll Rate (Sec): 2 Delta for Send: 0 Read Fails before Fault: 0

Timeout: 1 Max. Quiet Time (Sec): 0

Read Device Write Device OBJECT\_ANALOG\_INPUT #6 - read ok.

The following configuration parameters will appear when Map BACnet Object is checked.

Map BACnet Object	Check this box when the ValuPoint should look for another BACnet device to read data from or write data to.
Remote Device Instance	Enter the device instance of the remote BACnet device. The remote device must be able to support Who-Is/I-Am for locating device by device instance.
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'.
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.
Object Type	Select the object type from the list, such as Analog Input, etc.

Instance	Select the object instance. There are usually multiple instances of objects in a device. This identifies which one you want.
Data Type	Select data type 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. Data Type selection is only required when writing. Data is interpreted automatically when reading.
MAC Address	Sets the MS/TP MAC address (station ID) of the remote device if it is a slave-only device or does not support dynamic binding (Who-Is/I-Am). Normally you would leave this set to zero so that the remote device will be discovered automatically when the gateway broadcasts a "Who-Is" message. This setting applies to MS/TP only. BACnet IP devices (on the other side of a router) are required to support dynamic binding (Who-Is/I-Am).
Array Index	Some properties are an array of values. If applicable, enter the array index here. Enter -1 (negative one) to indicate no array index is to be used. Zero is a valid index.
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.
Poll Rate	This rate applies to periodic reading or writing, and simply specifies how often the remote device is read from or written to. (This parameter does not apply to physical I/O.)
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.
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.)
Timeout	When communicating with another remote device, the timeout given here specifies the amount of time the ValuPoint will wait for a response before calling it an error. NOTE: BACnet APDU timeout and retries are applied automatically independent of this timeout. Therefore, to be fully valid, this timeout should be Retries * APDU Timeout plus a margin. This timeout simply determines when the local object will be placed in the fault state with a non-zero reliability code.



Max 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 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.
----------------	--

Be sure to set the Priority when writing to a Commandable object. Failure to do so will result in the request being rejected. Conversely, be sure to select "None" when writing to a non-commandable object. Including priority for an object that does not expect it will also result in the request being rejected.

NOTE: There is specific behavior that applies when a commandable object (e.g. Analog Output) is writing to another commandable object as illustrated above. In this instance, priority 7 will be used to set the Present Value of the remote object as long as the local object is not in the relinquished state. If no priorities are set in the local object, then NULL will be written to priority 7 in the remote object effectively relinquishing that priority in the remote object.

The following screen shot illustrates an object configured to write its value to the remote BACnet device when the local object's value has changed by 1.

The screenshot shows the 'Object Map' tab in the VP6-1430 Configuration Tool v6.04. The interface is for configuring an object to write its value to a remote BACnet device. Key settings include:

- Object Type/Instance:** Analog Value, 2
- Object Name:** Analog Value 2
- Description:** Test client write
- Units:** square\_meters
- Write Settings:**
  - Write on Delta
  - Enable Max Quiet Time
  - Write Periodic
  - Set Default on Power-Up
  - Set Default on Comm Fail
  - Object is Persistent
- Remote Device Instance:** 40162
- Property:** Present Value
- Instance:** 2
- Data Type:** Real
- Priority:** None
- Delta for Send:** 1
- Max. Quiet Time (Sec):** 0

Buttons at the bottom include 'Read Device', 'Write Device', and a status bar showing 'OBJECT\_ANALOG\_INPUT #6 -- read ok.'

The following screen shot illustrates an object configured to read from a remote BACnet device and store the value received in the local object. Note that when reading, Data Type does not need to be set as the data type is included in the reply

form the remote device, and is automatically applied on the fly.

The screenshot displays the 'Object Map' configuration page in the 'VP6-1430 Configuration Tool v6.04'. The interface includes a toolbar with icons for file operations and a status bar showing 'Connected: [checked]', 'Target: [checked] 20801: ValuPoint VP6-1430', and 'In Sync: [checked]'. The main configuration area is divided into several sections:

- Object Type/Instance:** 'Analog Value' (dropdown), '3' (input).
- Object Name:** 'Analog Value 3' (input).
- Description:** 'Test object B' (input).
- Units:** 'square\_meters' (dropdown).
- Behavioral Options:**
  - Read Periodic
  - Write Periodic
  - Set Default on Power-Up
  - Enable Max Quite Time
  - Write on Delta
  - Set Default on Comm Fail
  - Object is Persistent
- COV and Scaling:**
  - Initial COV Increment: 0
  - Slope/Scale Factor: 0
  - Initial Relinquish Default: [input]
  - Initial COV Period: 0
  - Intercept/Offset: 0
  - Default Value: 0
- Mapping Options:**
  - Map Physical I/O
  - Map BACnet Object
  - Global Object
- Remote Device Instance:** 20841
- Property:** Present Value (dropdown), 85 (input)
- Bit Num:** 0 (input)
- Object Type:** Analog Input (dropdown)
- Instance:** 1 (input)
- Data Type:** Null (dropdown)
- Mac address (no who-is):** 201 (input)
- Array Index:** -1 (input)
- Priority:** None (dropdown)
- Poll Rate (Sec):** 2 (input)
- Delta for Send:** 0 (input)
- Read Fails before Fault:** 0 (input)
- Timeout:** 0 (input)
- Max. Quiet Time (Sec):** 0 (input)

At the bottom, there are 'Read Device' and 'Write Device' buttons, and a status field showing 'OBJECT\_ANALOG\_INPUT #6 - read ok.'

The following screen shot shows an example of what the Object List page will look like with a few objects configured to communicate with a remote BACnet device.

VP6-1430 Configuration Tool v6.05 [vp6-1430-2.csv]

Connected:  Target:  20801: ValuPoint VP6-1430 In Sync:

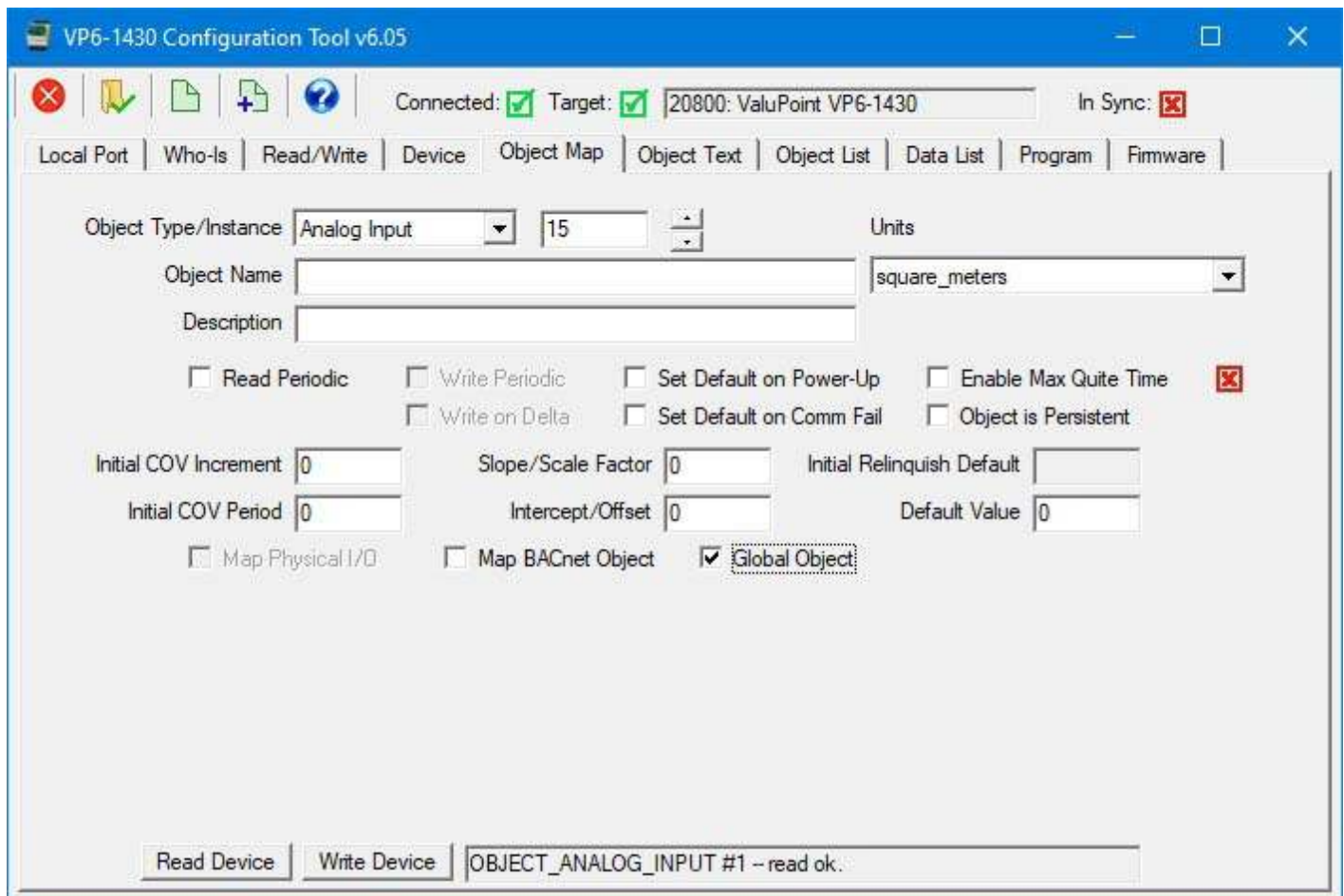
Local Port | Who-Is | Read/Write | Device | Object Map | Object Text | Object List | Data List | Program | Firmware

Read All Write All

Default BACnet Device: 1

Obj...	Name	R/W	Device	Object	Property	Configuration
AI 2	No name [AI 2]	---	Local	AI [2]	A/UI #2	0-10V, 12-bit
AI 3	No name [AI 3]	---	Local	AI [3]	A/UI #3	0-10V, 12-bit
AI 4	No name [AI 4]	---	Local	AI [4]	A/UI #4	0-10V, 12-bit
AI 5	No name [AI 5]	---	Local	AI [5]	A/UI #5	0-10V, 12-bit
AI 6	No name [AI 6]	---	Local	AI [6]	A/UI #6	0-10V, 12-bit
AI 7	No name [AI 7]	---	Local	AI [7]	A/UI #7	0-10V, 12-bit
AI 8	No name [AI 8]	---	Local	AI [8]	A/UI #8	0-10V, 12-bit
AI 9	No name [AI 9]	---	Local	AI [9]	A/UI #9	0-10V, 12-bit
AI 10	No name [AI 10]	---	Local	AI [10]	A/UI #10	0-10V, 12-bit
AI 11	No name [AI 11]	---	Local	AI [11]	A/UI #11	0-10V, 12-bit
AI 12	No name [AI 12]	---	Local	AI [12]	A/UI #12	0-10V, 12-bit
AI 13	Remote Item 1	R	MS/TP [123]	AI [1]	Present_Value	
AI 14	Remote Item 2	R	MS/TP [123]	AI [2]	Present_Value	
AI 15	Remote Item 3	R	MS/TP [123]	AI [3]	Present_Value	
AI 16	Remote Item 4	R	MS/TP [123]	AI [4]	Present_Value	
AI 17		---	None	None	None	
AI 18		---	None	None	None	

## 7.4 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 i.CanDrawIt 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.

## 7.5 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.

## 7.6 Export and Import Configuration with XML File

Export and import on this page work exactly the same as XML file handling on the Device page. Refer to Section 6.7 for a detailed discussion of file export and import.

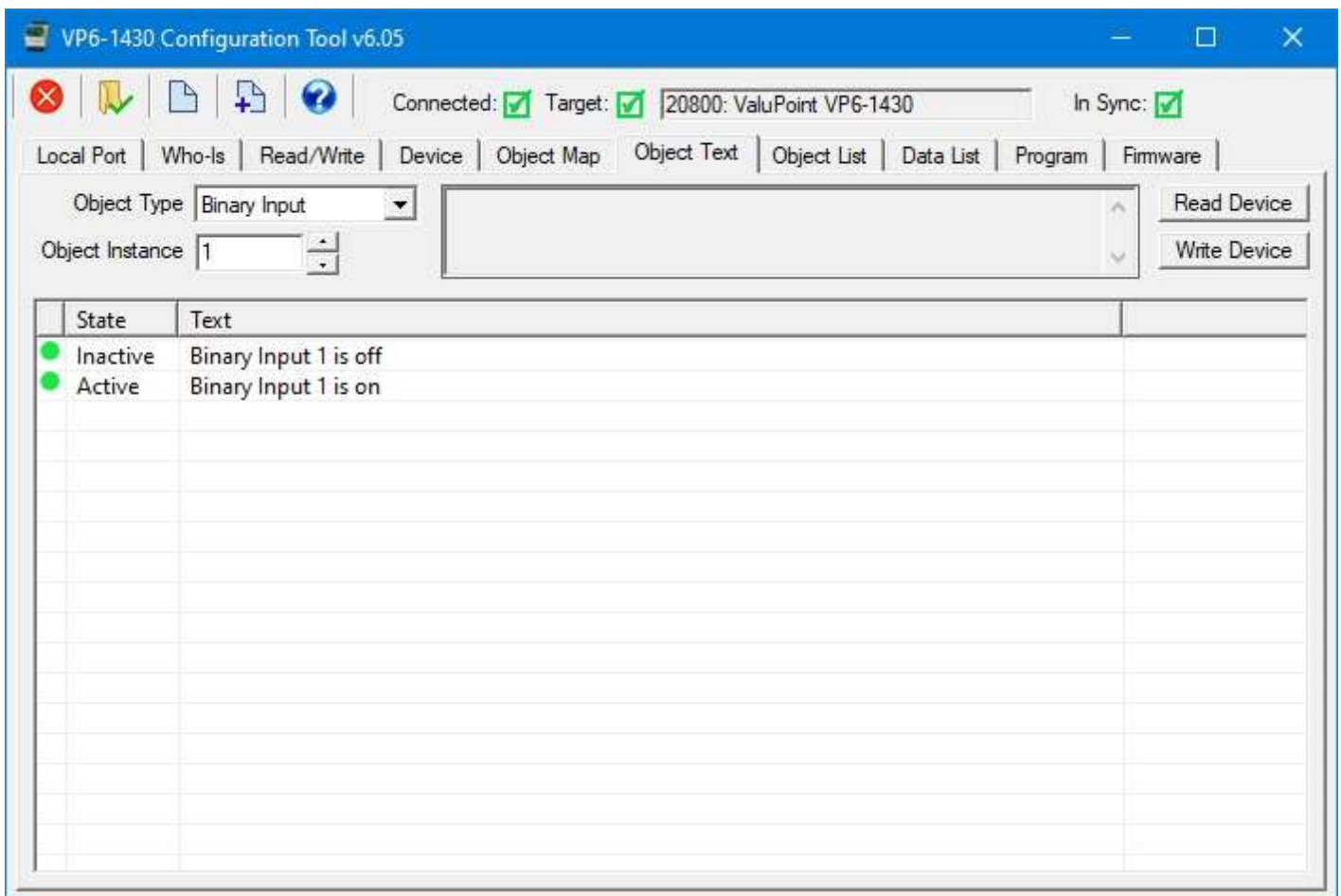


## 8 Tool 'Object Text' Page

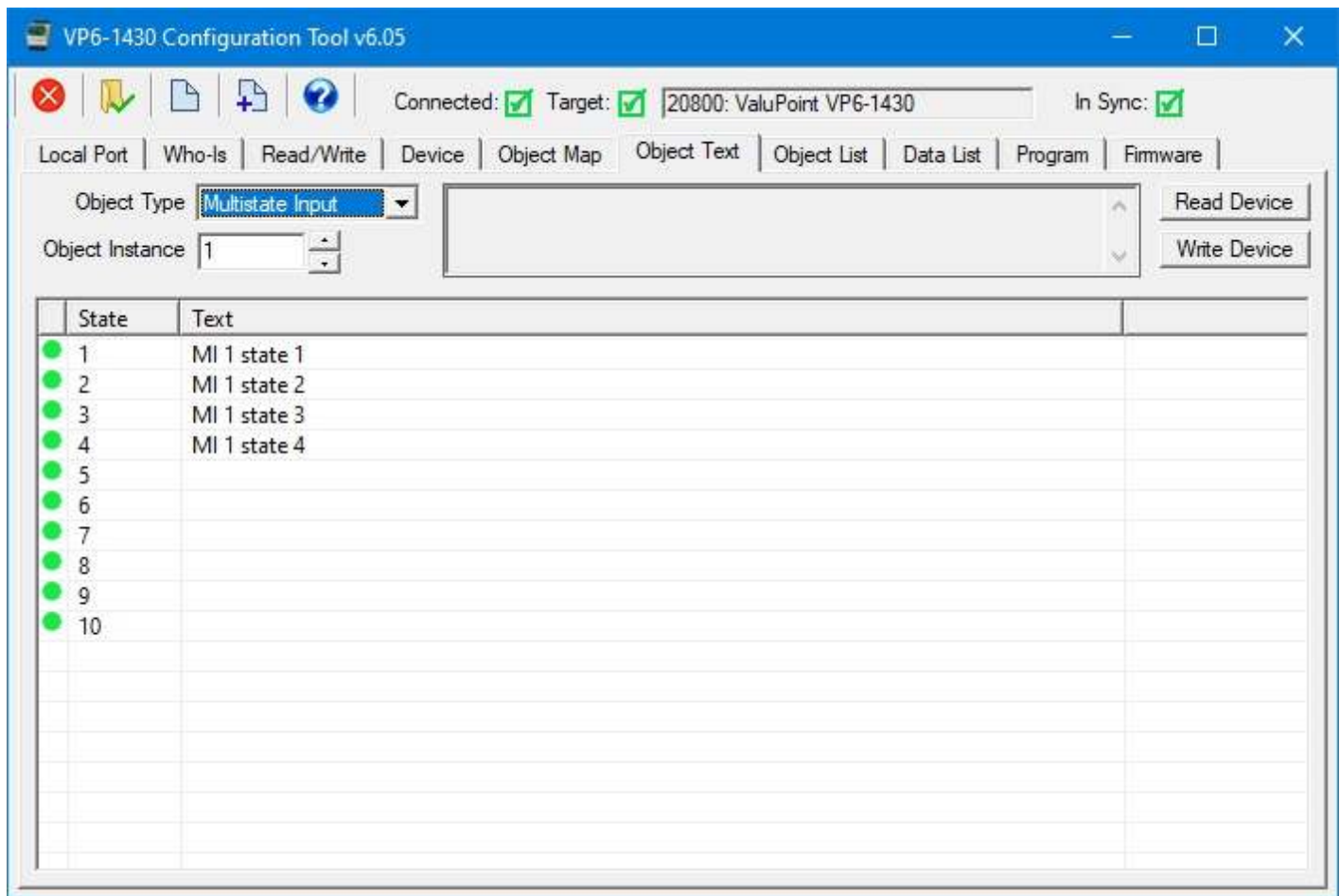
Binary objects have an active and inactive state string associated with them. Multistate objects have state strings with one string per state. This page is where you edit those strings. Using state strings is not required, but if you do use them, this page is where you may edit them.

### 8.1 Read/Write Object Text

Select object type and instance, then click Read Device to retrieve the strings from the gateway. For Binary objects, there will be only two states labeled Inactive and Active.

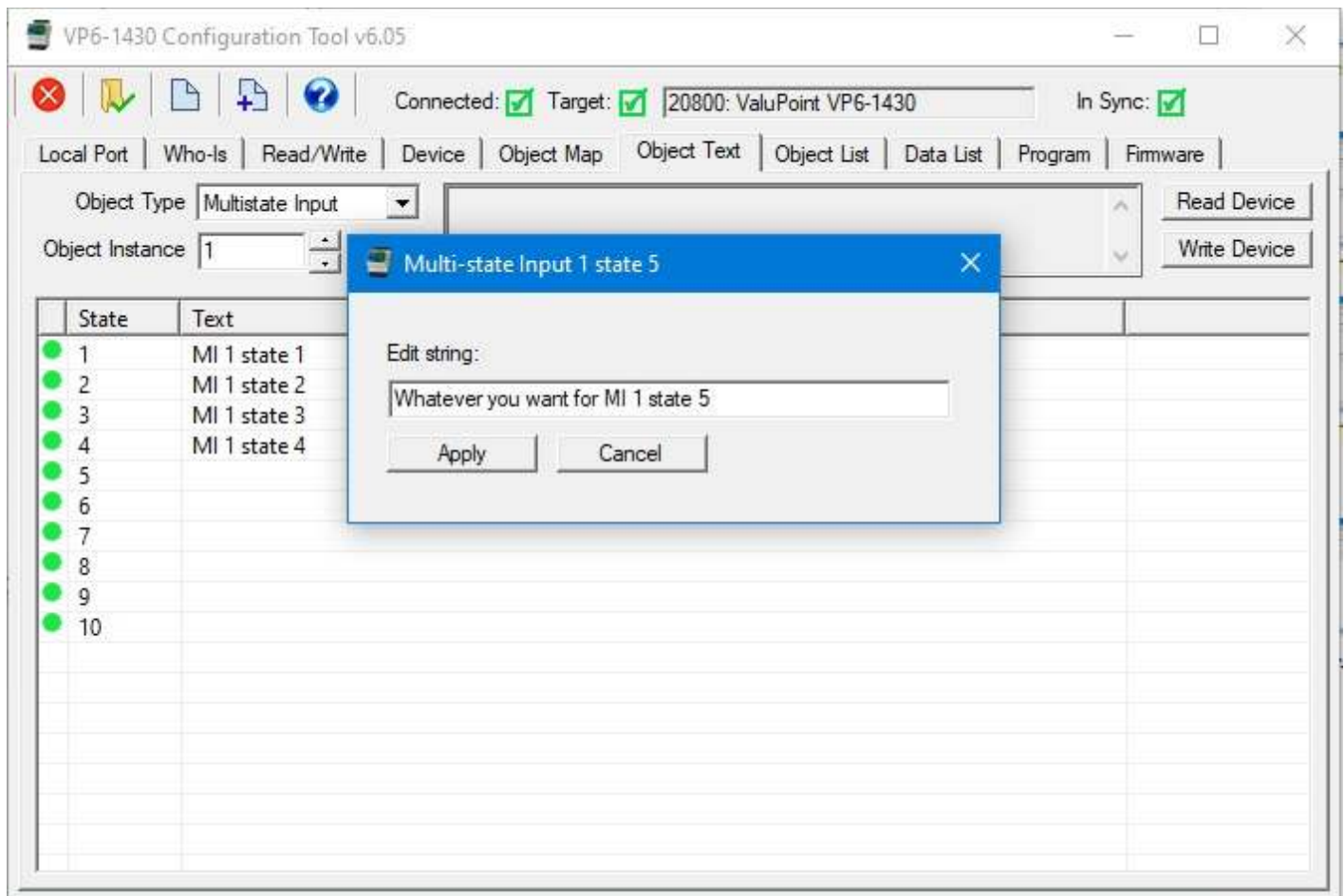


When retrieving state strings for a Multistate object, there will be one string per state up to the maximum state set on the Device page at the bottom of the object count list.



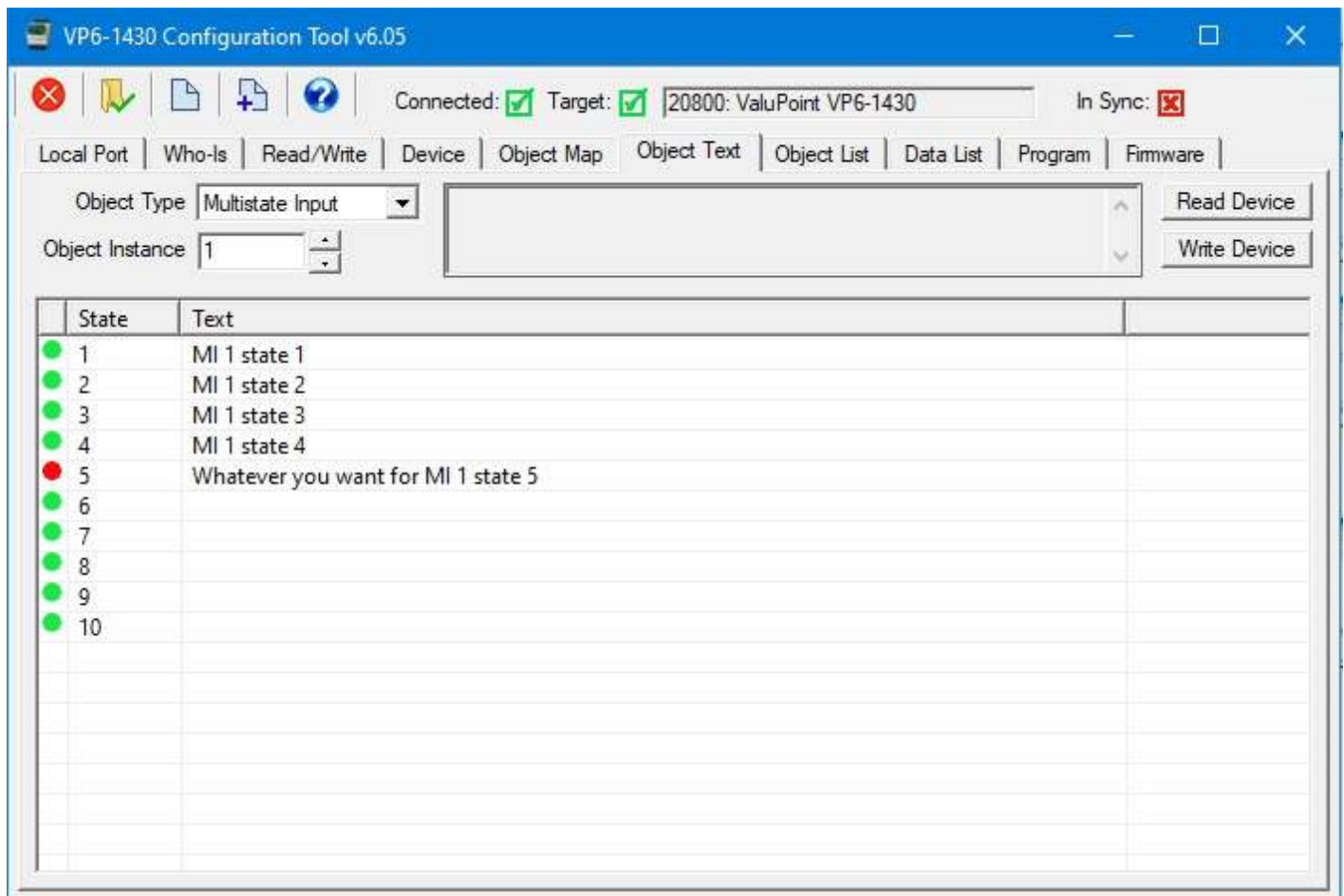
## 8.2 Edit Object Text

Double click on a line of text to open the edit dialog. You may edit or enter a new text string. Click Apply to apply the new or modified string.



After making changes to text strings, the icon will change from green to red reminding you that you still need to send the text to the gateway by clicking Write Device.

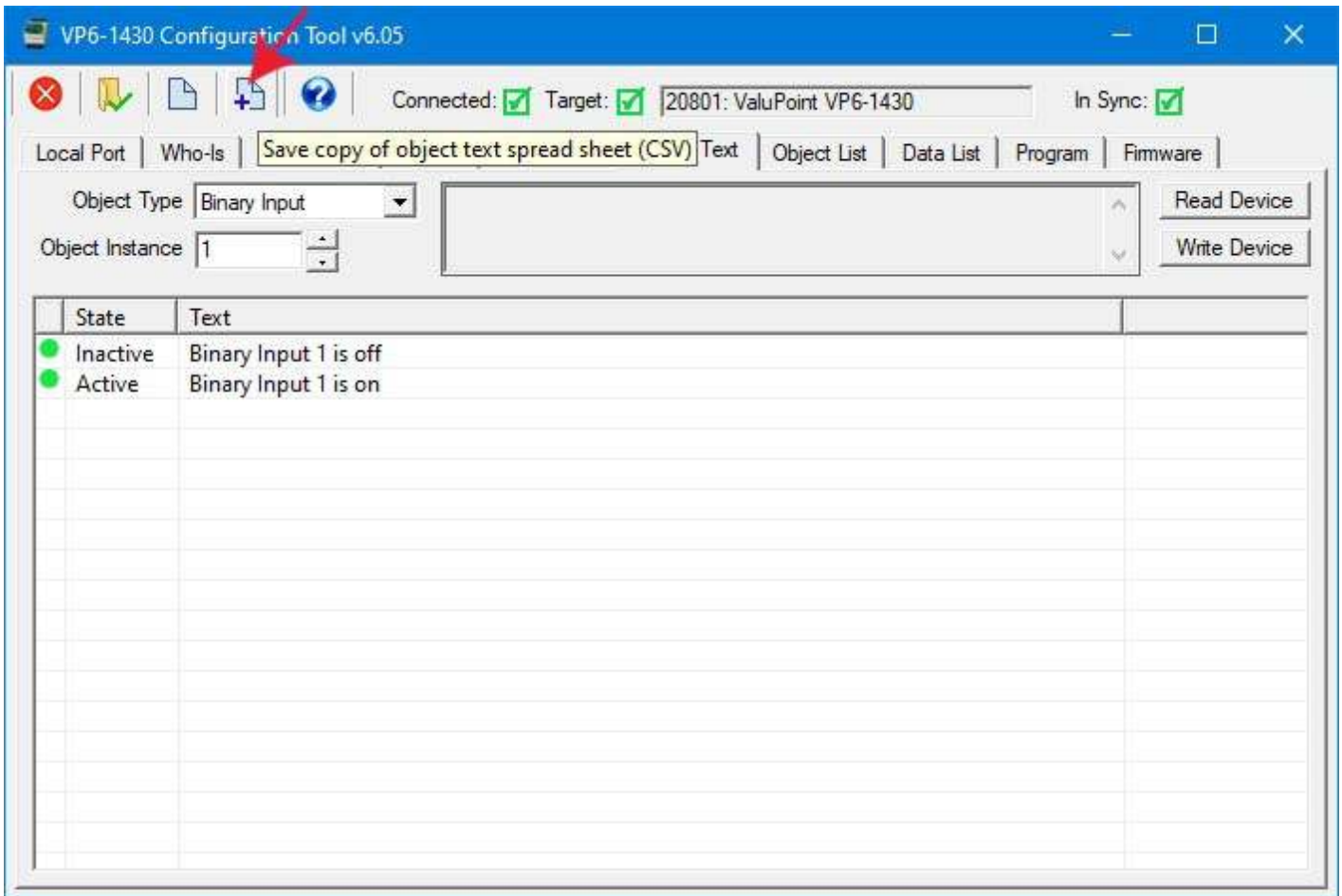




### 8.3 Export and Import Text as CSV file

Text strings may be exported and imported using a simple CSV file having only 3 columns: Object, state, and text. For Binary objects, the state should be 0=inactive, 1=active. For Multistate objects, states should begin with 1 and count up to the maximum value configured on the Device page.

Click on the Save icon to save the current text values to a CSV file. The usual Windows dialog for selecting a folder and file name will appear. Proceed as you would for saving any other sort of file on your PC.



You can now open the CSV file in any spread sheet program.

File Home Insert Page Layout Formulas Data Review View Help

Cut Copy Paste Format Painter Clipboard

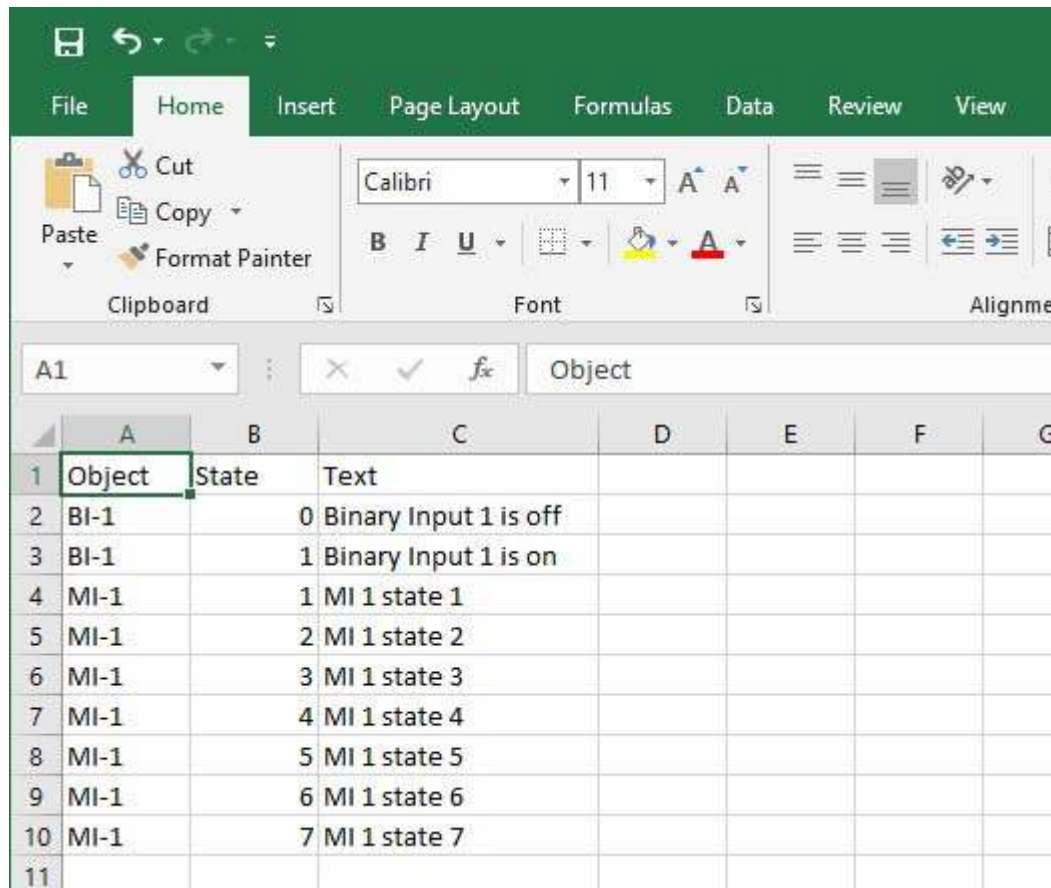
Calibri 11 Font

Alignment

A1 Object

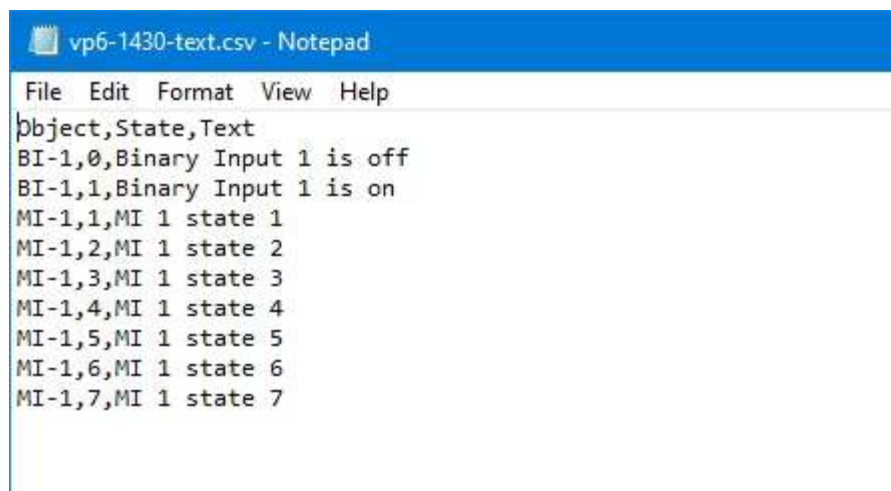
Object	State	Text
BI-1	0	Binary Input 1 is off
BI-1	1	Binary Input 1 is on
MI-1	1	MI 1 state 1
MI-1	2	MI 1 state 2
MI-1	3	MI 1 state 3
MI-1	4	MI 1 state 4

Edit the file as you wish to add strings.



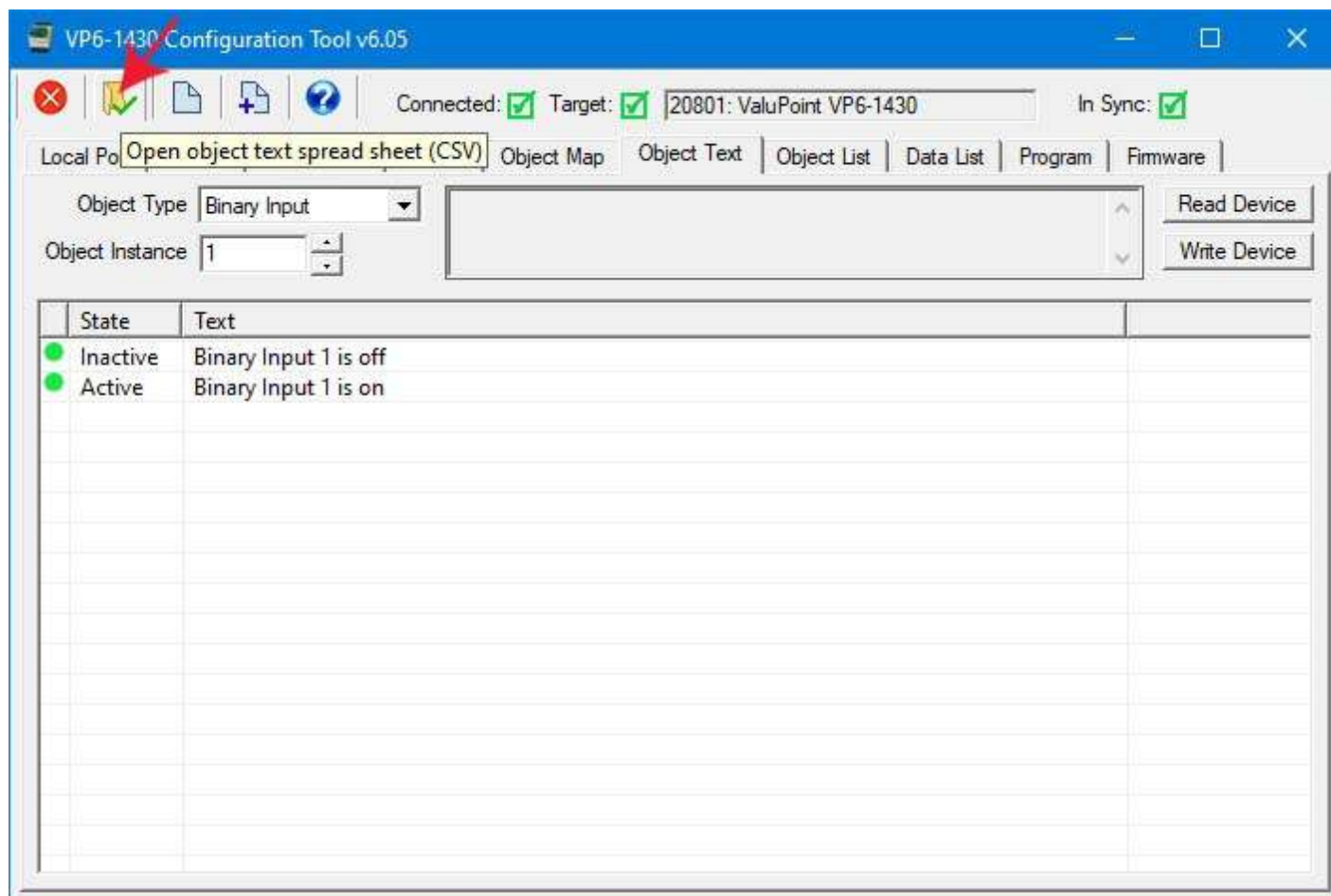
	A	B	C	D	E	F	G
1	Object	State	Text				
2	BI-1	0	Binary Input 1 is off				
3	BI-1	1	Binary Input 1 is on				
4	MI-1	1	MI 1 state 1				
5	MI-1	2	MI 1 state 2				
6	MI-1	3	MI 1 state 3				
7	MI-1	4	MI 1 state 4				
8	MI-1	5	MI 1 state 5				
9	MI-1	6	MI 1 state 6				
10	MI-1	7	MI 1 state 7				
11							

The text view of the same CSV file would appear as follows. You can use a simple text editor if you are careful to preserve the CSV format.

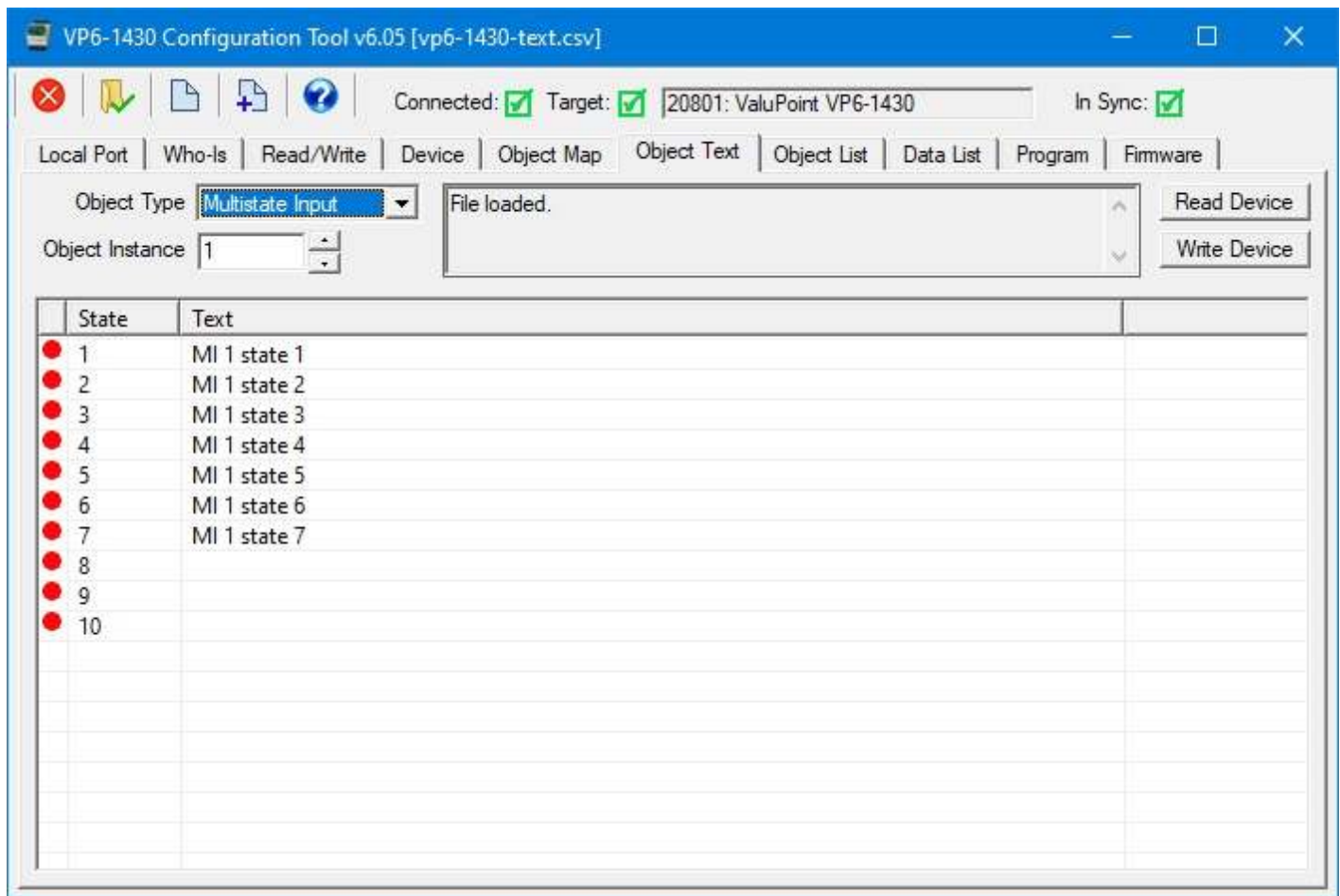


```
vp6-1430-text.csv - Notepad
File Edit Format View Help
Object,State,Text
BI-1,0,Binary Input 1 is off
BI-1,1,Binary Input 1 is on
MI-1,1,MI 1 state 1
MI-1,2,MI 1 state 2
MI-1,3,MI 1 state 3
MI-1,4,MI 1 state 4
MI-1,5,MI 1 state 5
MI-1,6,MI 1 state 6
MI-1,7,MI 1 state 7
```

Click on the Open icon to import your file back into the configuration tool.



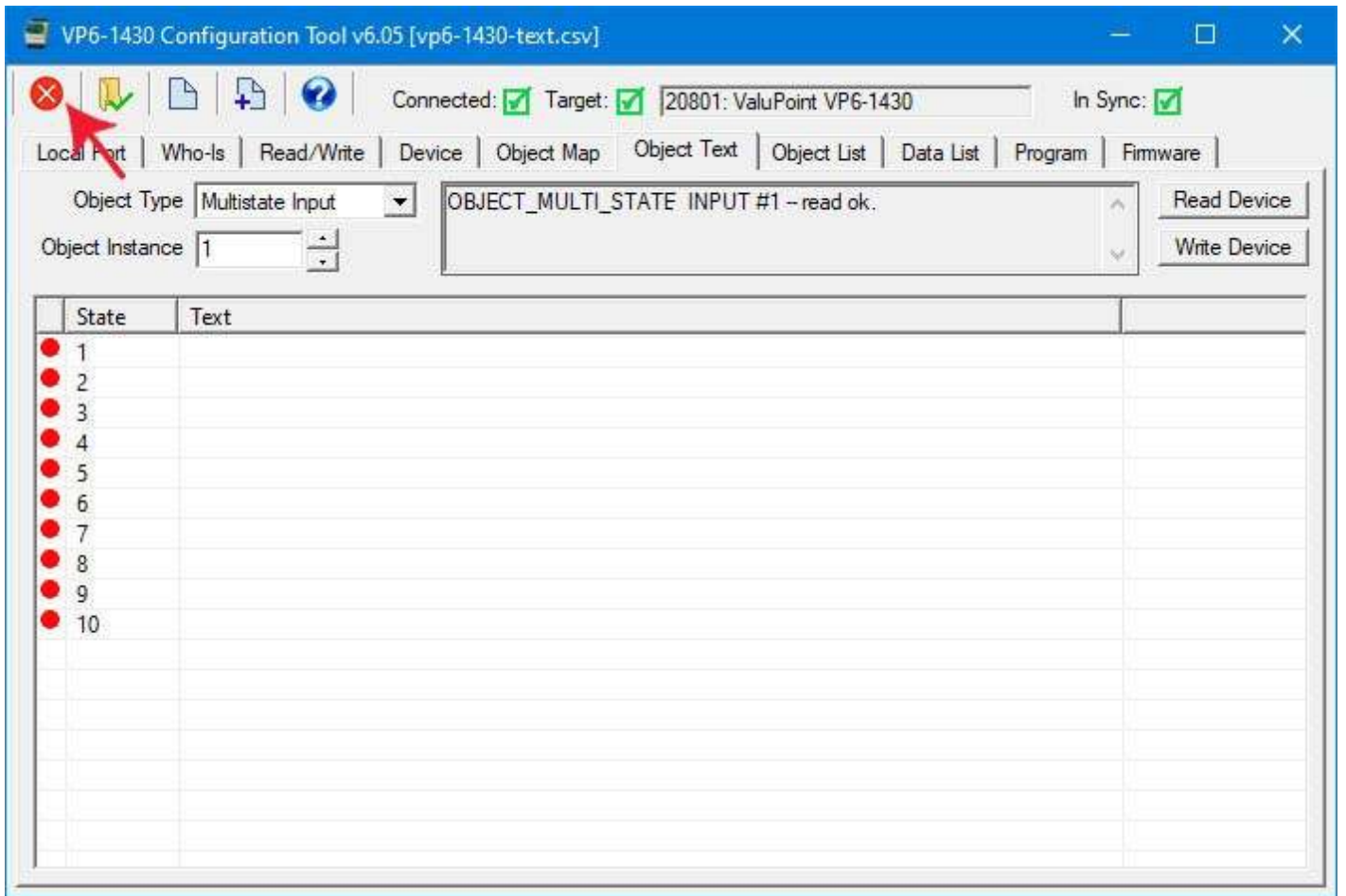
Upon loading the CSV file, the text strings will appear on the Object Text page. You may click Write Device to write text strings one object at a time, or go to the Object List page and click Write All.



## 8.4 Clearing Object Text

Clicking the Clear icon (first icon on the left in the top toolbar) on the Object Text page only clears the object text. It does not affect object properties, object counts or device properties. If you want to clear any of those, go to the Object Map or Device pages and click the Clear icon there to clear. The Clear icon on the Object Text page clears text for all objects having text properties.

The buttons in the configuration tool only clear information stored by the tool software in your PC's memory. To completely clear configuration in the physical gateway device, you need to follow the "Reconfigure Object Counts" procedure discussed in Section 6 of this user guide. To clear just object text, use the Write Device button on the Object Text page to clear just one object, or use the Write All button on the Object List page.





## 9 Object List Page

The Object List page is where you can import and export object mappings in CSV file format. This is more convenient for making mass changes to the configuration.

### 9.1 Read/Write All Object Maps in Device

You first need to confirm that you have some objects allocated on the Device page. Click Get Counts to retrieve the currently configured count of objects in the device. Refer to Section 6 of this user guide for more about object counts.

The Object List page will initially appear largely empty, except that the fixed I/O assignments will appear in the list.

VP6-1430 Configuration Tool v6.05

Connected:  Target:  20801: ValuPoint VP6-1430 In Sync:

Local Port | Who-Is | Read/Write | Device | Object Map | Object Text | Object List | Data List | Program | Firmware

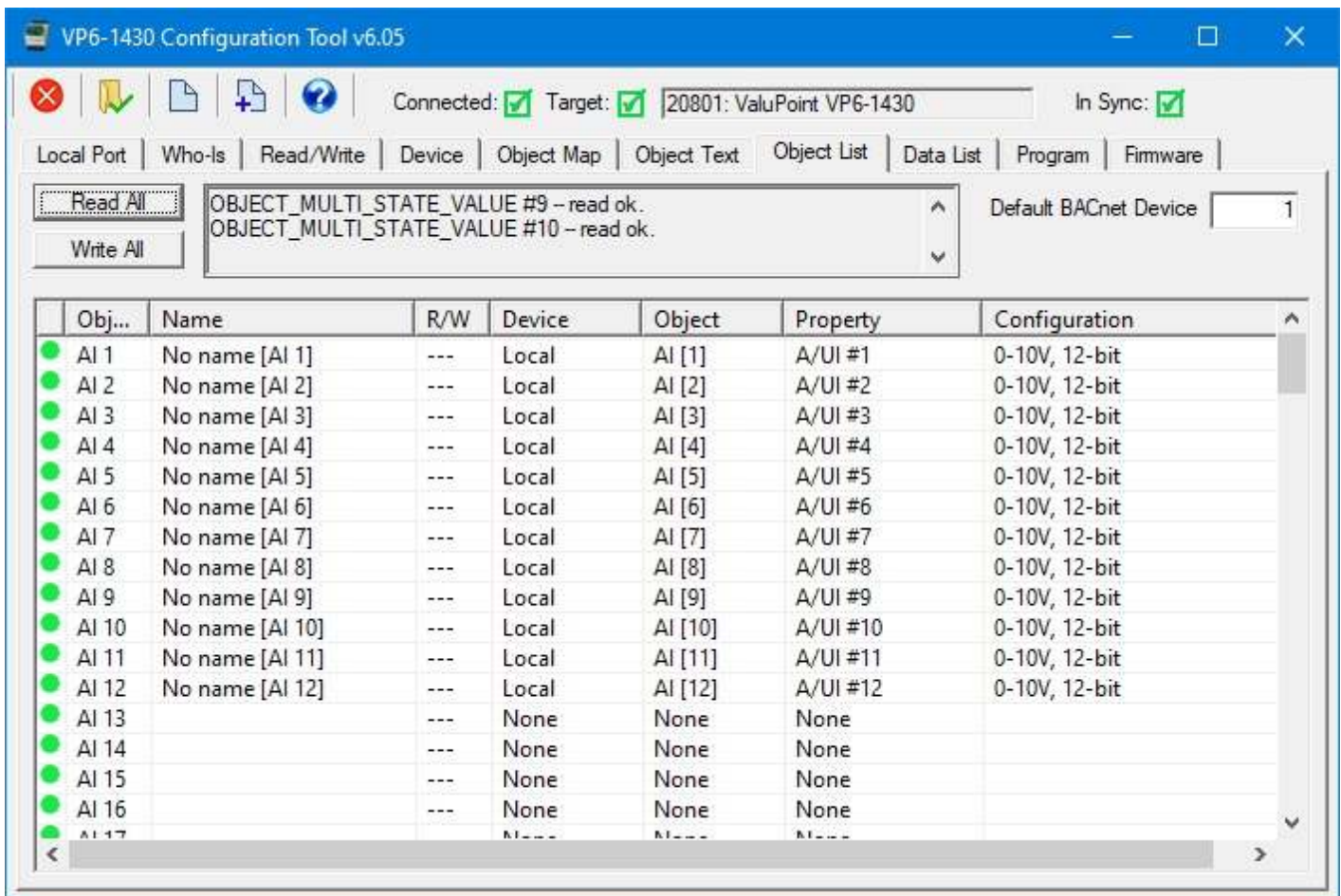
Read All Write All

Default BACnet Device

Obj...	Name	R/W	Device	Object	Property	Configuration
AI 1		---	Local	AI [1]	A/UI #1	Unspecified/Unused
AI 2		---	Local	AI [2]	A/UI #2	Unspecified/Unused
AI 3		---	Local	AI [3]	A/UI #3	Unspecified/Unused
AI 4		---	Local	AI [4]	A/UI #4	Unspecified/Unused
AI 5		---	Local	AI [5]	A/UI #5	Unspecified/Unused
AI 6		---	Local	AI [6]	A/UI #6	Unspecified/Unused
AI 7		---	Local	AI [7]	A/UI #7	Unspecified/Unused
AI 8		---	Local	AI [8]	A/UI #8	Unspecified/Unused
AI 9		---	Local	AI [9]	A/UI #9	Unspecified/Unused
AI 10		---	Local	AI [10]	A/UI #10	Unspecified/Unused
AI 11		---	Local	AI [11]	A/UI #11	Unspecified/Unused
AI 12		---	Local	AI [12]	A/UI #12	Unspecified/Unused
AI 13		---	None	None	None	
AI 14		---	None	None	None	
AI 15		---	None	None	None	
AI 16		---	None	None	None	
AI 17		---	None	None	None	

Click Read All to begin the process of reading all object maps from the device. As objects are read, they will be populated in the object list, and the in-sync icon will

change from red to green to indicate that this is the information actually stored in the device. Until the icon turns green, any configuration information displayed is a copy that only exists in the configuration tool software on your PC.



As the process continues, the "read ok" status lines will continue to scroll up in the status window. When this activity stops, it is generally safe to assume everything has been read. But the icons in the first column will confirm that. If green, then that line contains information actually read from the device.

When you make changes to object maps, the green in-sync icon will change to red. To write the updated configuration to the device, click the Write All button. As the process continues, the "write ok" will be written to the status window and the in-sync icons will turn from red to green.

## 9.2 Select an Object Map for Editing

You cannot make changes to object maps from the Object List page. To make changes, double click the line you wish to change. You will be automatically taken to the Object Map page where that object will be displayed. You may make changes there.



Obj...	Name	R/W	Device	Object	Property	Configuration
AI 1	No name [AI 1]	---	Local	AI [1]	A/UI #1	4-20mA, with resistor
AI 2	No name [AI 2]	---	Local	AI [2]	A/UI #2	0-10V, 12-bit
AI 3	No name [AI 3]	---	Local	AI [3]	A/UI #3	0-10V, 12-bit
AI 4	No name [AI 4]	---	Local	AI [4]	A/UI #4	0-10V, 12-bit
AI 5	No name [AI 5]	---	Local	AI [5]	A/UI #5	0-10V, 12-bit
AI 6	No name [AI 6]	---	Local	AI [6]	A/UI #6	0-10V, 12-bit
AI 7	No name [AI 7]	---	Local	AI [7]	A/UI #7	0-10V, 12-bit
AI 8	No name [AI 8]	---	Local	AI [8]	A/UI #8	0-10V, 12-bit
AI 9	No name [AI 9]	---	Local	AI [9]	A/UI #9	0-10V, 12-bit
AI 10	No name [AI 10]	---	Local	AI [10]	A/UI #10	0-10V, 12-bit
AI 11	No name [AI 11]	---	Local	AI [11]	A/UI #11	0-10V, 12-bit
AI 12	No name [AI 12]	---	Local	AI [12]	A/UI #12	0-10V, 12-bit
AI 13		---	None	None	None	
AI 14		---	None	None	None	
AI 15		---	None	None	None	
AI 16		---	None	None	None	
AI 17		---	None	None	None	

### 9.3 Export and Import Configuration as CSV file

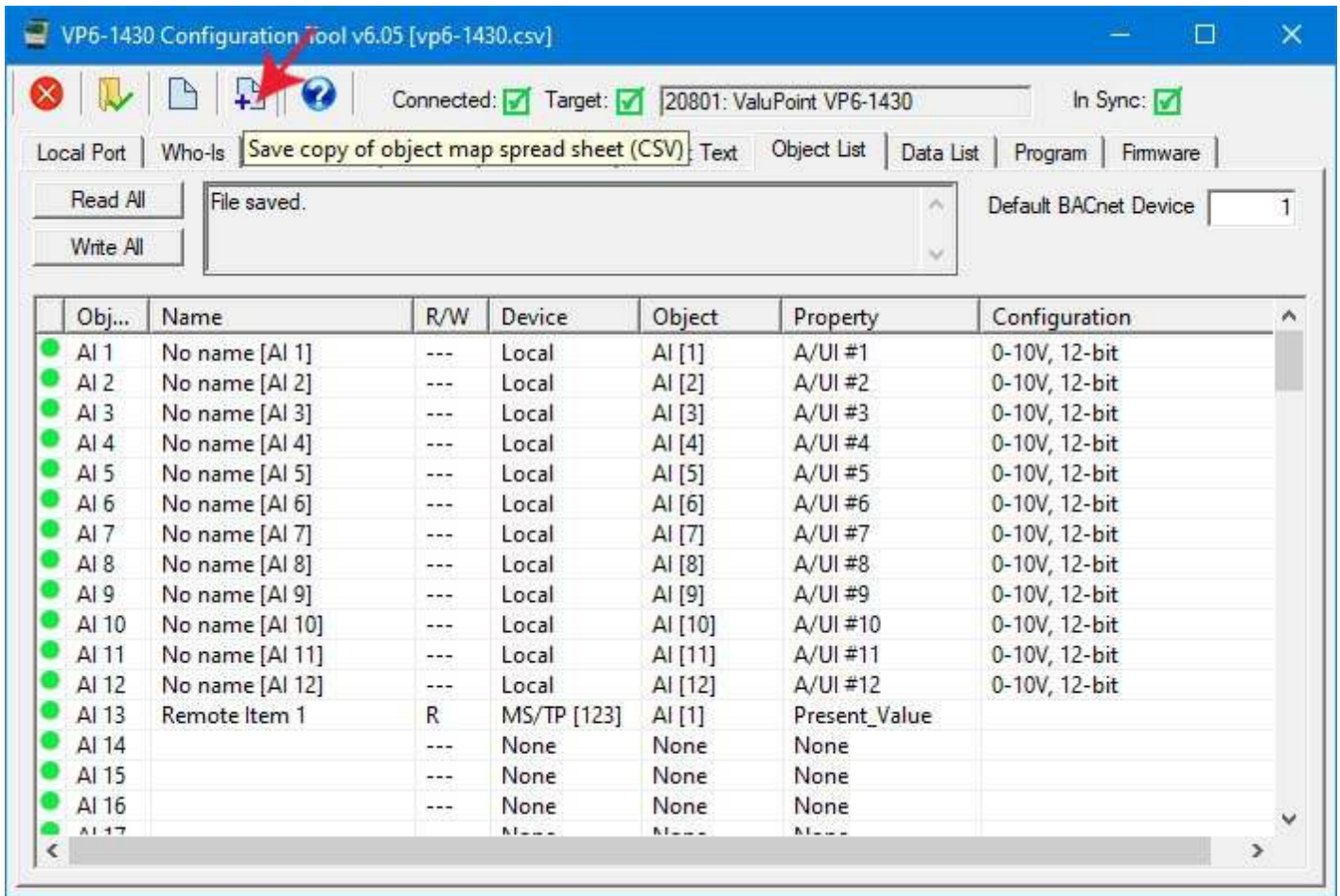
To create a long list of object maps that will communicate with remote BACnet devices, it is most useful to go to the Object Map page and create one mapping as a template. You can then copy/paste or "fill down" to replicate this map and make only minor changes.

The screenshot shows the 'VP6-1430 Configuration Tool v6.05 [vp6-1430.csv]' window. The 'Object Map' tab is selected. The interface includes a toolbar with icons for Local Port, Who-Is, Read/Write, Device, Object Map, Object Text, Object List, Data List, Program, and Firmware. The main configuration area is divided into several sections:

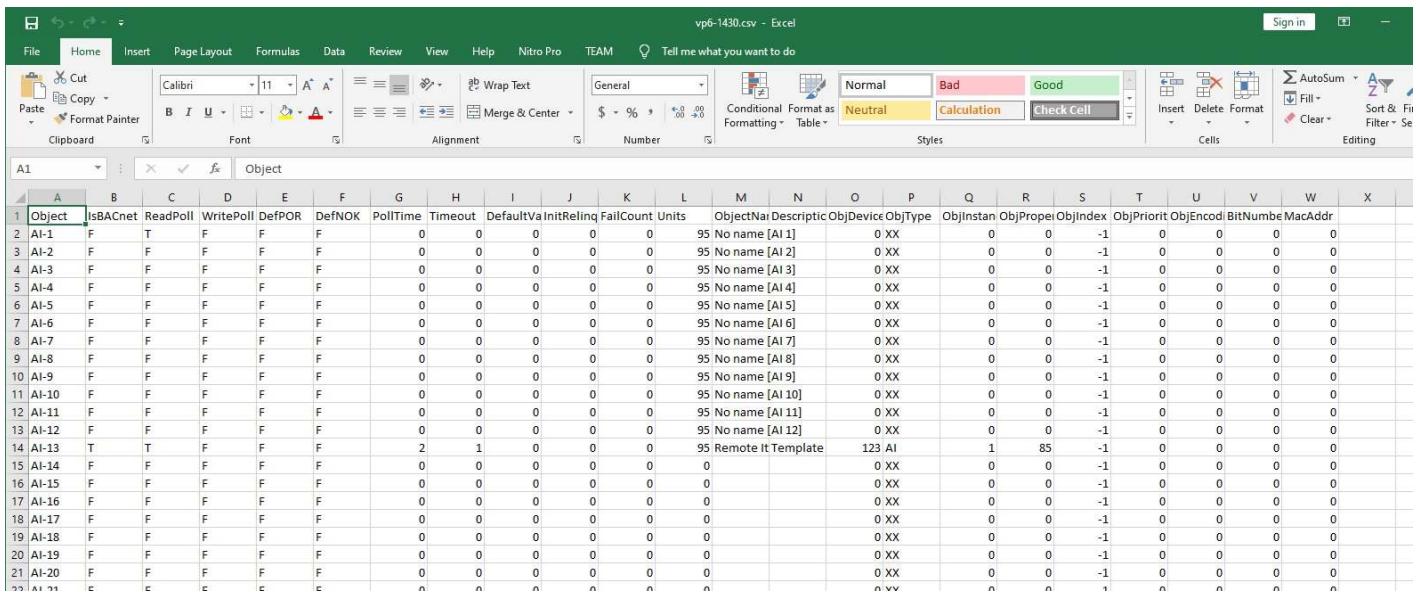
- Object Type/Instance:** Analog Input (dropdown), 13 (input), Units: no\_units (dropdown).
- Object Name:** Remote Item 1 (input), Units: no\_units (dropdown).
- Description:** Template for client (input).
- Checkboxes:**
  - Read Periodic
  - Write Periodic
  - Set Default on Power-Up
  - Enable Max Quiet Time
  - Write on Delta
  - Set Default on Comm Fail
  - Object is Persistent
- COV Parameters:**
  - Initial COV Increment: 0 (input)
  - Slope/Scale Factor: 0 (input)
  - Initial Relinquish Default: (input)
  - Initial COV Period: 0 (input)
  - Intercept/Offset: 0 (input)
  - Default Value: 0 (input)
- Mapping Options:**
  - Map Physical I/O
  - Map BACnet Object
  - Global Object
- Remote Device Instance:** 123 (input)
- Property:** Present Value (dropdown), 85 (input), Bit Num: 0 (input)
- Object Type:** Analog Input (dropdown), Instance: 1 (input), Data Type: Null (dropdown)
- Mac address (no who-is):** 0 (input), Array Index: -1 (input), Priority: None (dropdown)
- Poll Rate (Sec):** 2 (input), Delta for Send: 0 (input), Read Fails before Fault: 0 (input)
- Timeout:** 1 (input), Max. Quiet Time (Sec): 0 (input)

At the bottom, there are buttons for 'Read Device' and 'Write Device'. The 'Write Device' button is highlighted, and the status bar shows 'OBJECT\_ANALOG\_INPUT #13 -- write ok.'

To save your object maps as a CSV file, click the New File icon. The Windows file dialog will appear, and you may then enter the name under which the file should be saved.



If you subsequently open the saved CSV file with a spread sheet program, it will appear something like this. Refer to Appendix C for a complete listing of all of the possible columns in the spread sheet. When saving a CSV file, the configuration tool will skip any columns that are not used at least once in your configuration.



In this example, we are replicating the template, and making minor modifications to each map.

Object	ISBACnet	ReadPoll	WritePoll	DefPOR	DefNOK	PollTime	Timeout	DefaultValue	InitRelinquish	FailCount	Units	ObjectName	Description	ObjDevice	ObjType	ObjInstan	ObjPropri	ObjIndex	ObjPriorit	ObjEncod	BITNumbe	MacAddr
AI-13	T	T	F	F	F	2	1	0	0	0	95	Remote Item 1	Client test item 1	123	AI	1	85	-1	0	0	0	0
AI-14	T	T	F	F	F	2	1	0	0	0	95	Remote Item 2	Client test item 2	123	AI	2	85	-1	0	0	0	0
AI-15	T	T	F	F	F	2	1	0	0	0	95	Remote Item 3	Client test item 3	123	AI	3	85	-1	0	0	0	0
AI-16	T	T	F	F	F	2	1	0	0	0	95	Remote Item 4	Client test item 4	123	AI	4	85	-1	0	0	0	0

The text representation of this CSV file will look like this:

```

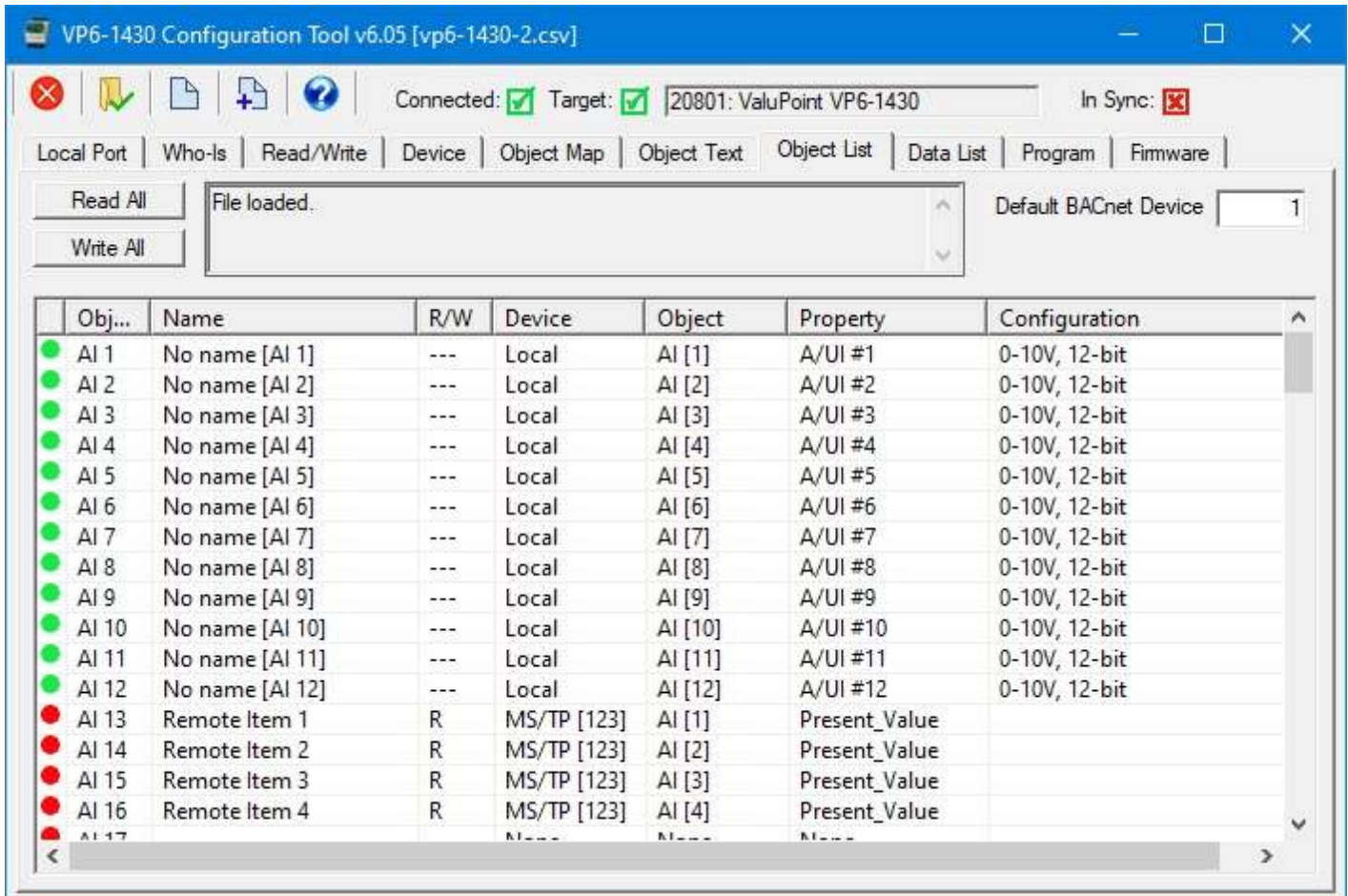
Object,IsBACnet,ReadPoll,WritePoll,DefPOR,DefNOK,PollTime,Timeout,DefaultValue,InitRelinquish,FailCount,Units,ObjectName,Desc
AI-13,T,T,F,F,2,1,0,0,0,95,Remote Item 1,Client test item 1,123,AI,1,85,-1,0,0,0,0
AI-14,T,T,F,F,2,1,0,0,0,95,Remote Item 2,Client test item 2,123,AI,2,85,-1,0,0,0,0
AI-15,T,T,F,F,2,1,0,0,0,95,Remote Item 3,Client test item 3,123,AI,3,85,-1,0,0,0,0
AI-16,T,T,F,F,2,1,0,0,0,95,Remote Item 4,Client test item 4,123,AI,4,85,-1,0,0,0,0

```

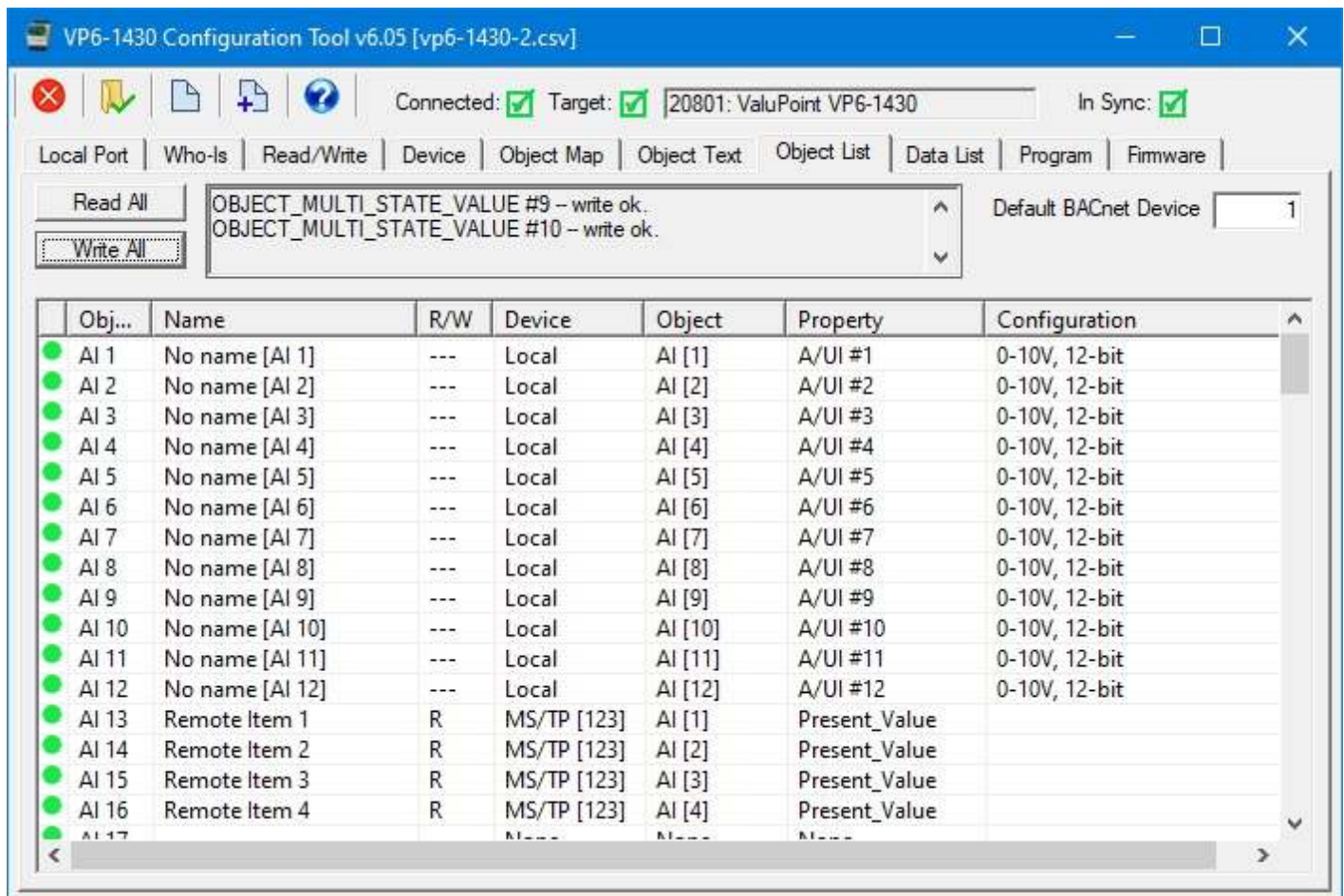
To clear the object maps and reload a previously saved CSV file, start by clicking the Clear icon (first icon on the left). Then click the File Open icon. The Windows file dialog will appear, allowing you to select the file to be loaded.

Obj...	Name	R/W	Device	Object	Property	Configuration
AI 1	No name [AI 1]	---	Local	AI [1]	A/UI #1	0-10V, 12-bit
AI 2	No name [AI 2]	---	Local	AI [2]	A/UI #2	0-10V, 12-bit
AI 3	No name [AI 3]	---	Local	AI [3]	A/UI #3	0-10V, 12-bit
AI 4	No name [AI 4]	---	Local	AI [4]	A/UI #4	0-10V, 12-bit
AI 5	No name [AI 5]	---	Local	AI [5]	A/UI #5	0-10V, 12-bit
AI 6	No name [AI 6]	---	Local	AI [6]	A/UI #6	0-10V, 12-bit
AI 7	No name [AI 7]	---	Local	AI [7]	A/UI #7	0-10V, 12-bit
AI 8	No name [AI 8]	---	Local	AI [8]	A/UI #8	0-10V, 12-bit
AI 9	No name [AI 9]	---	Local	AI [9]	A/UI #9	0-10V, 12-bit
AI 10	No name [AI 10]	---	Local	AI [10]	A/UI #10	0-10V, 12-bit
AI 11	No name [AI 11]	---	Local	AI [11]	A/UI #11	0-10V, 12-bit
AI 12	No name [AI 12]	---	Local	AI [12]	A/UI #12	0-10V, 12-bit
AI 13	Remote Item 1	R	MS/TP [123]	AI [1]	Present_Value	
AI 14		---	None	None	None	
AI 15		---	None	None	None	
AI 16		---	None	None	None	

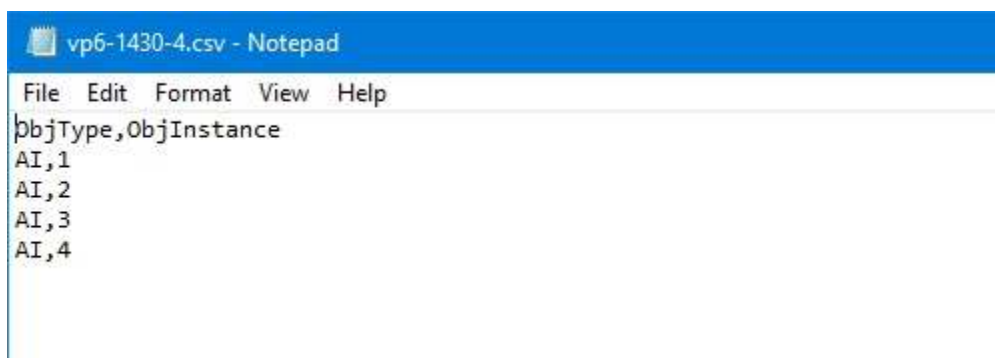
Upon loading the CSV file, the screen looks somewhat like it did when we first saved the file above, except the icons in the first column are red instead of green. Green means the object map is in sync with the device. Red means the object map contained within the configuration tool is not in sync with object maps actually contained within the device itself. You need to click Write All to write the maps to the device, and then they will become in sync.



Once the newly loaded maps are written to the device, the icons will become green.



It is not necessary to include all columns when importing a CSV file. The import utility is intended to make configuration of a list of objects as easy as possible. The minimum requirement for importing a list of objects to read from a remote device is simply the object type and number you wish to read from that remote device. The device instance will take on the default value indicated in the upper right corner of the Object List page when importing. The first available local object will be assigned, and Present Value is the property assumed to be read since this is the most common.



The very simply CSV file pictured above will result in the import illustrated below, assuming objects starting at AI 13 had not been previously used.

Obj...	Name	R/W	Device	Object	Property	Configuration
AI 1	No name [AI 1]	---	Local	AI [1]	A/UI #1	4-20mA, with resistor
AI 2	No name [AI 2]	---	Local	AI [2]	A/UI #2	0-10V, 12-bit
AI 3	No name [AI 3]	---	Local	AI [3]	A/UI #3	0-10V, 12-bit
AI 4	No name [AI 4]	---	Local	AI [4]	A/UI #4	0-10V, 12-bit
AI 5	No name [AI 5]	---	Local	AI [5]	A/UI #5	0-10V, 12-bit
AI 6	No name [AI 6]	---	Local	AI [6]	A/UI #6	0-10V, 12-bit
AI 7	No name [AI 7]	---	Local	AI [7]	A/UI #7	0-10V, 12-bit
AI 8	No name [AI 8]	---	Local	AI [8]	A/UI #8	0-10V, 12-bit
AI 9	No name [AI 9]	---	Local	AI [9]	A/UI #9	0-10V, 12-bit
AI 10	No name [AI 10]	---	Local	AI [10]	A/UI #10	0-10V, 12-bit
AI 11	No name [AI 11]	---	Local	AI [11]	A/UI #11	0-10V, 12-bit
AI 12	No name [AI 12]	---	Local	AI [12]	A/UI #12	0-10V, 12-bit
AI 13	AI-13	R	MS/TP [201]	AI [1]	Present_Value	
AI 14	AI-14	R	MS/TP [201]	AI [2]	Present_Value	
AI 15	AI-15	R	MS/TP [201]	AI [3]	Present_Value	
AI 16	AI-16	R	MS/TP [201]	AI [4]	Present_Value	
AI 17						

## 9.4 Clearing Object Maps, Counts, and Device

Clicking the Clear icon (first icon on the left in the top toolbar) on the Object List page only clears the object maps. It does not affect object counts or device properties. If you want to change any of those, go to the Device page and click the Clear icon there to clear.

The buttons in the configuration tool only clear information stored by the tool software in your PC's memory. To clear configuration in the physical device, you need to follow the "Reconfigure Object Counts" procedure discussed in Section 6 of this user guide.

VP6-1430 Configuration Tool v6.05

Connected:  Target:  20801: ValuPoint VP6-1430 In Sync:

Local Port | Who-Is | Read/Write | Device | Object Map | Object Text | **Object List** | Data List | Program | Firmware

Read All Write All

Default BACnet Device: 1

Obj...	Name	R/W	Device	Object	Property	Configuration
● AI 1		---	Local	AI [1]	A/UI #1	Unspecified/Unused
● AI 2		---	Local	AI [2]	A/UI #2	Unspecified/Unused
● AI 3		---	Local	AI [3]	A/UI #3	Unspecified/Unused
● AI 4		---	Local	AI [4]	A/UI #4	Unspecified/Unused
● AI 5		---	Local	AI [5]	A/UI #5	Unspecified/Unused
● AI 6		---	Local	AI [6]	A/UI #6	Unspecified/Unused
● AI 7		---	Local	AI [7]	A/UI #7	Unspecified/Unused
● AI 8		---	Local	AI [8]	A/UI #8	Unspecified/Unused
● AI 9		---	Local	AI [9]	A/UI #9	Unspecified/Unused
● AI 10		---	Local	AI [10]	A/UI #10	Unspecified/Unused
● AI 11		---	Local	AI [11]	A/UI #11	Unspecified/Unused
● AI 12		---	Local	AI [12]	A/UI #12	Unspecified/Unused
● AI 13		---	None	None	None	
● AI 14		---	None	None	None	
● AI 15		---	None	None	None	
● AI 16		---	None	None	None	
● AI 17		---	None	None	None	





## 10 Tool 'Data List' Page

### 10.1 Read All Object Data

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.

To manually change the values in any BACnet object in the gateway device, use the Read/Write page to access object properties.

Obj...	R/W	Device	Object	Property	Present Value	Reliabil...	Status Flags
AI 7	---	Local	AI [7]	A/UI #7	0.002785	0	F,F,F,F { }
AI 8	---	Local	AI [8]	A/UI #8	0.000000	0	F,F,F,F { }
AI 9	---	Local	AI [9]	A/UI #9	0.002785	0	F,F,F,F { }
AI 10	---	Local	AI [10]	A/UI #10	0.000000	0	F,F,F,F { }
AI 11	---	Local	AI [11]	A/UI #11	0.000000	0	F,F,F,F { }
AI 12	---	Local	AI [12]	A/UI #12	0.000000	0	F,F,F,F { }
AO 1	---	None	None	None	123.000000	0	F,F,F,F { }
AV 1	R	MS/TP [40162]	AV [1]	Present_Value	11.550000	0	F,F,F,F { }
AV 2	W+	MS/TP [40162]	AV [2]	Present_Value	457.100006	0	F,F,F,F { }
AV 3	R	MS/TP [20841]	AI [1]	Present_Value	0.000000	0	F,F,F,F { }
AV 4	R	MS/TP [16010]	AI [1]	Present_Value	0.000000	0	F,F,F,F { }
BI 1	---	Local	BI [1]	A/UI #1	0	0	F,F,F,F { }
BI 2	---	Local	BI [2]	A/UI #2	0	0	F,F,F,F { }
BI 3	---	Local	BI [3]	A/UI #3	0	0	F,F,F,F { }
BI 4	---	Local	BI [4]	A/UI #4	0	0	F,F,F,F { }
BI 5	---	Local	BI [5]	A/UI #5	0	0	F,F,F,F { }
BI 6	---	Local	BI [6]	A/UI #6	0	0	F,F,F,F { }

Obj...  
(Object)

The first "Obj..." column is the local object identification.

R/W	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. The R/W designation only applies to objects that are mapped to communicate with remote BACnet devices.
Device	The Device column will show what remote device this object is mapped to. 'MS/TP[N]' means BACnet slave whose device instance is N.
Object	The Object column shows what local object is mapped to physical I/O when the Device is "Local". The Object column shows what object in the remote device is mapped to this local object when communicating with a remote BACnet device. For a BACnet object, the object type and instance are shown, e.g. 'AI[1]' means Analog Input 1. In the case of a commandable object, you will see something like "AO[2] P=3" - this means Analog Output 2, and write priority is set to 3.
Property	The Property column will show the assigned I/O point for "Local" objects, or the BACnet object property being read when the object is mapped to read remote devices.
Present Value	Present Value shows a representation of the present value of the BACnet object in the VP6-1430.
Reliability	Reliability code for the BACnet object in the VP6-1430 is shown. These are listed below.
Status Flags	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.

## 10.2 Error Codes

If the remote device being queried by the VP6-1430 returns an error code, or there is simply a problem in communicating with the remote device, this will be indicated by the local object's reliability code. When reliability code is non-zero, the fault flag is also set. Therefore, the Status Flags indication will typically be "F,T,F,F" any time the reliability code is something other than zero (zero means no errors to report).

Error codes returned by a remote BACnet device are encoded into reliability codes presented by the respective object. 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).

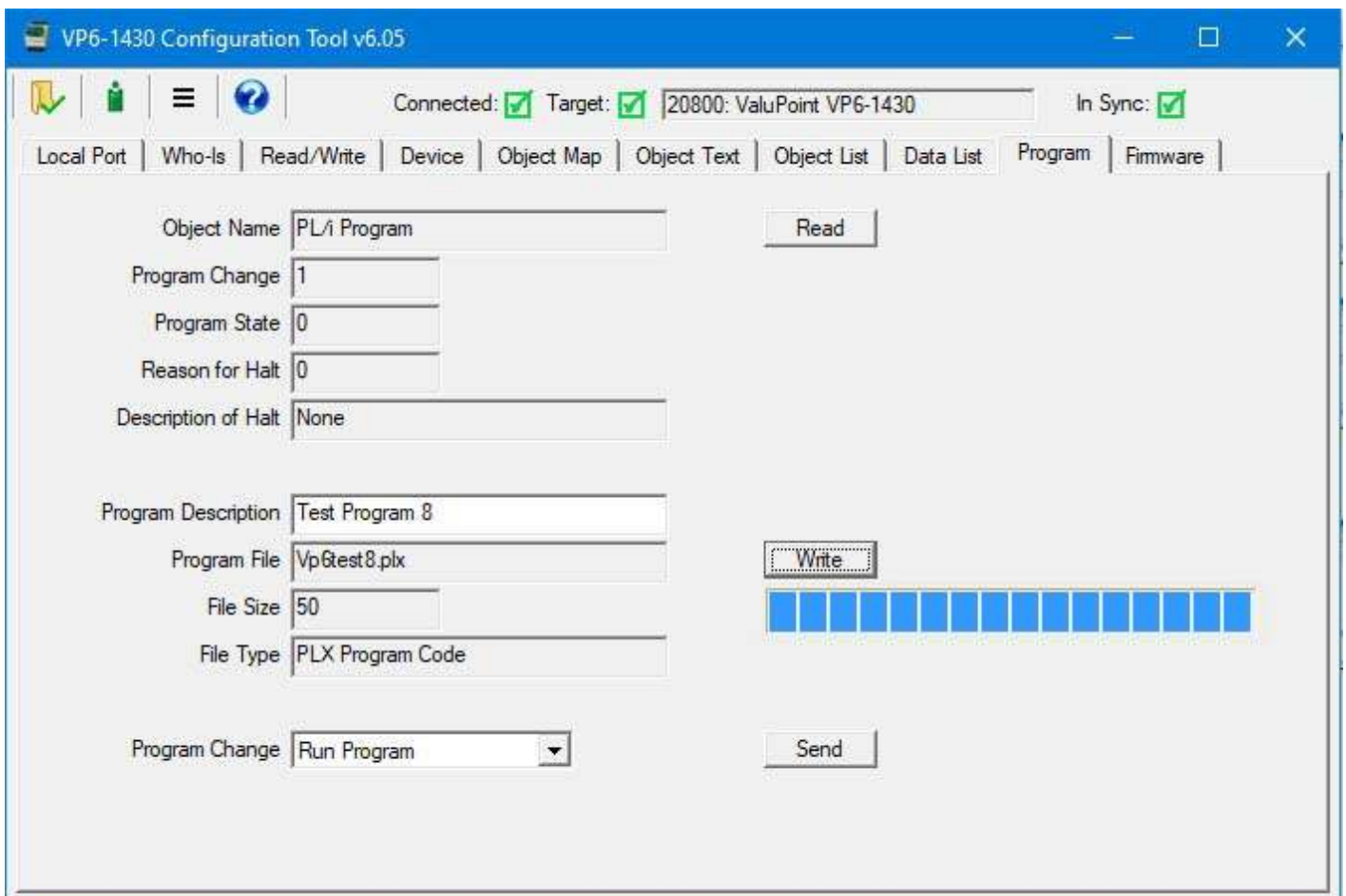


## 11 Programming Page

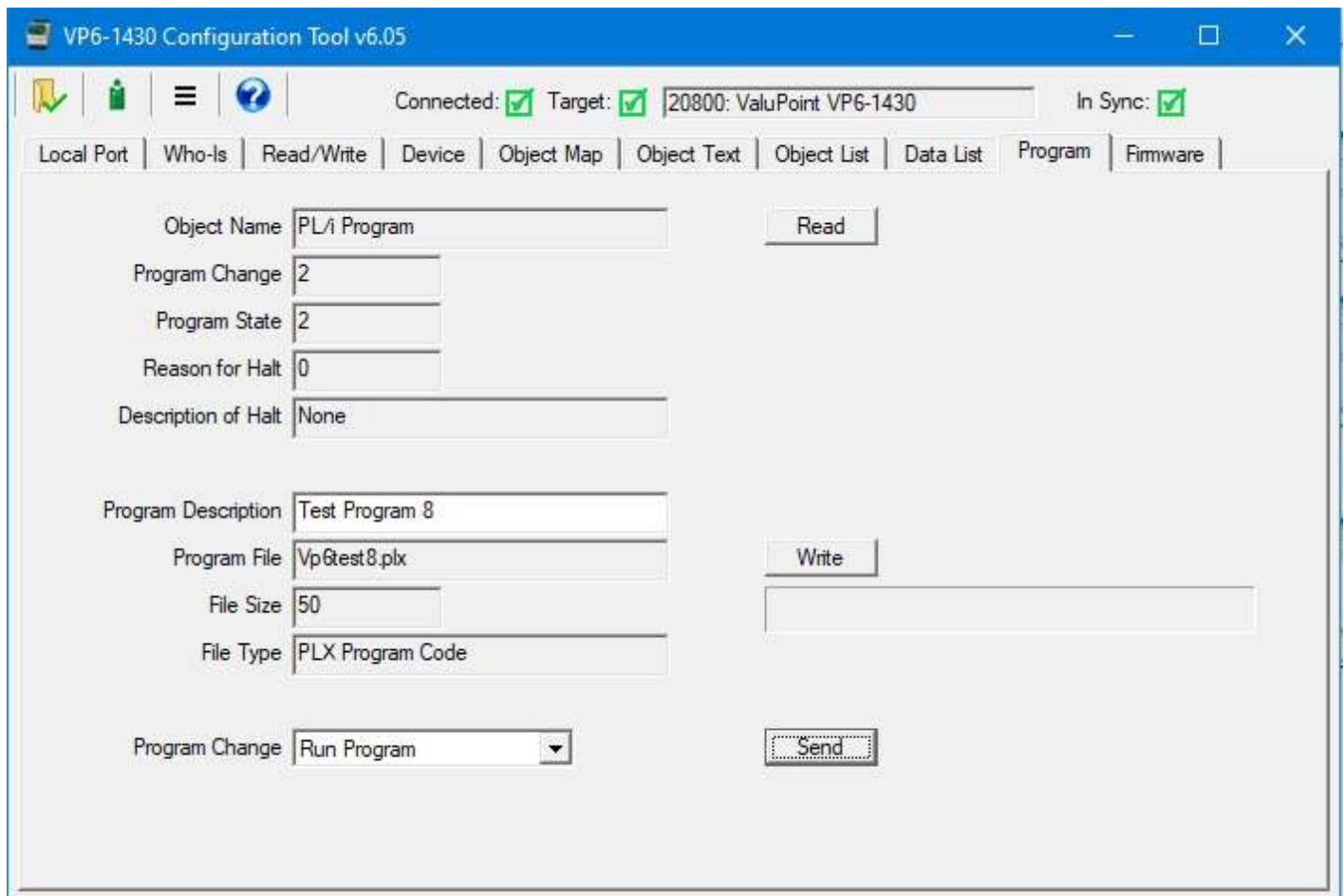
### 11.1 Program Loading and Execution

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.

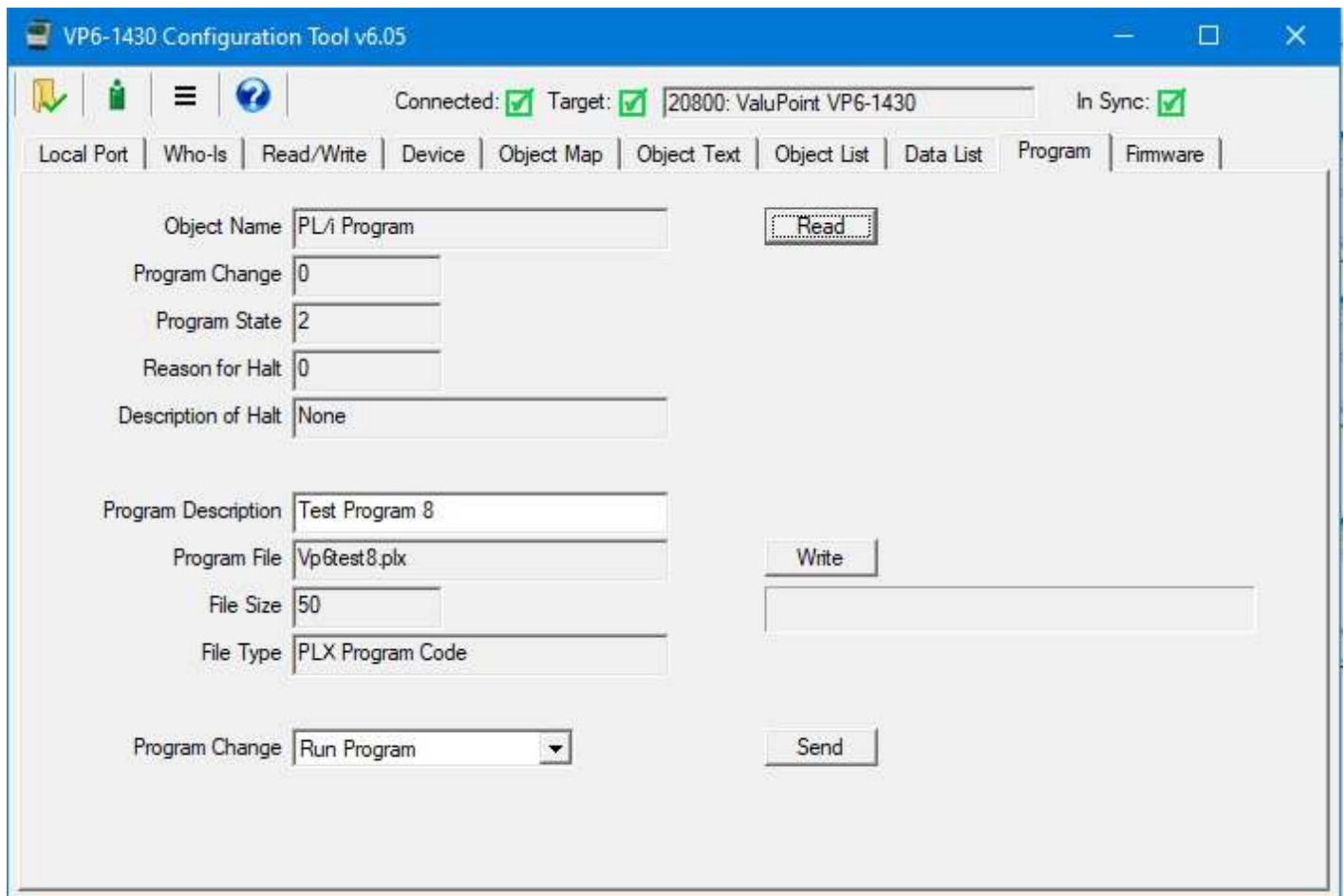
After a program (.plx file) has been opened, click the Upload button to send that program to the ValuPoint. A progress bar will indicate program loading progress.



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

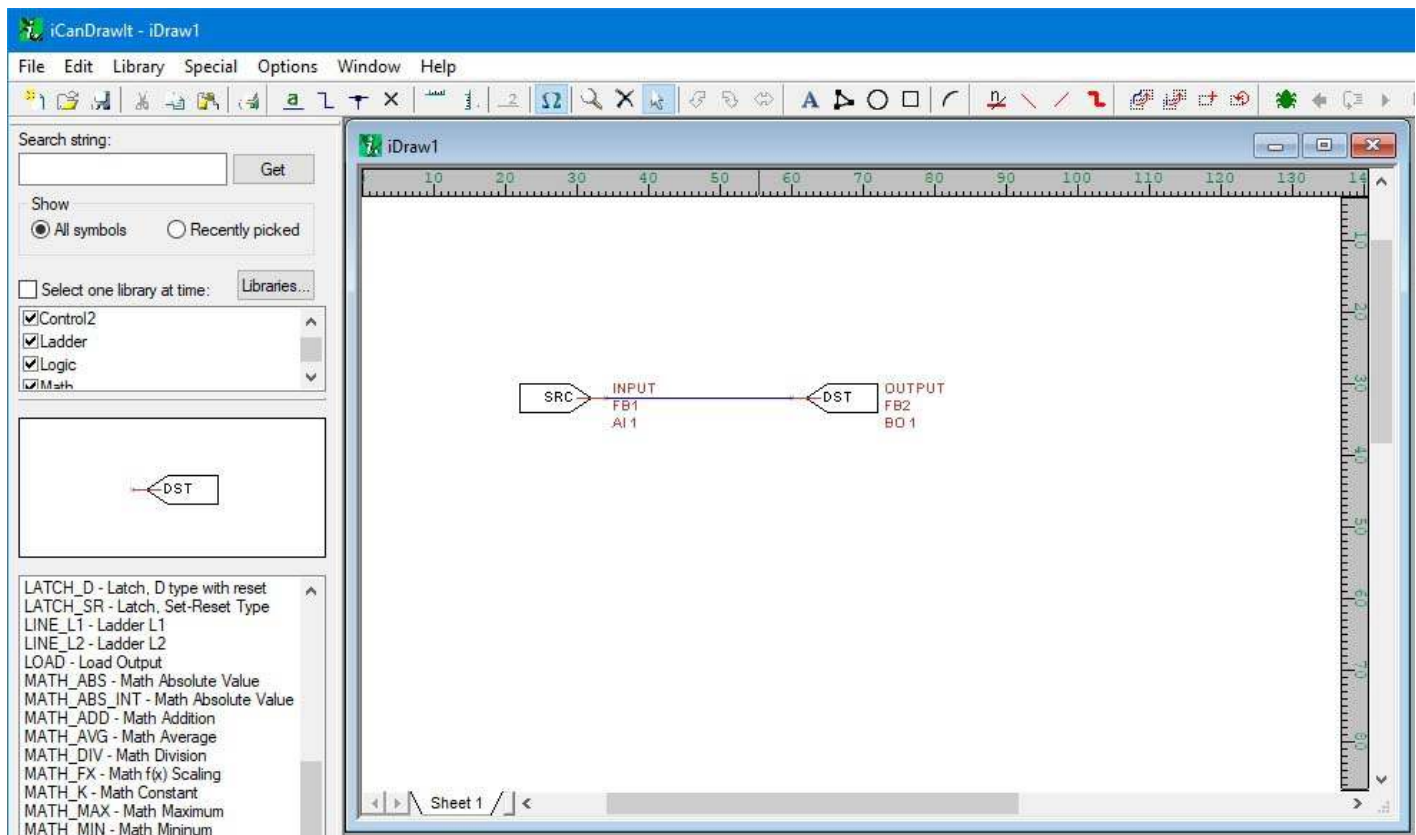


To stop the program, select Halt Program and click Request. To check the status of the program, just click Read.



If the program encounters a fatal error during execution, its error code and description of halt will be displayed after clicking the Read button.

## 11.2 Program Editing and Debugging

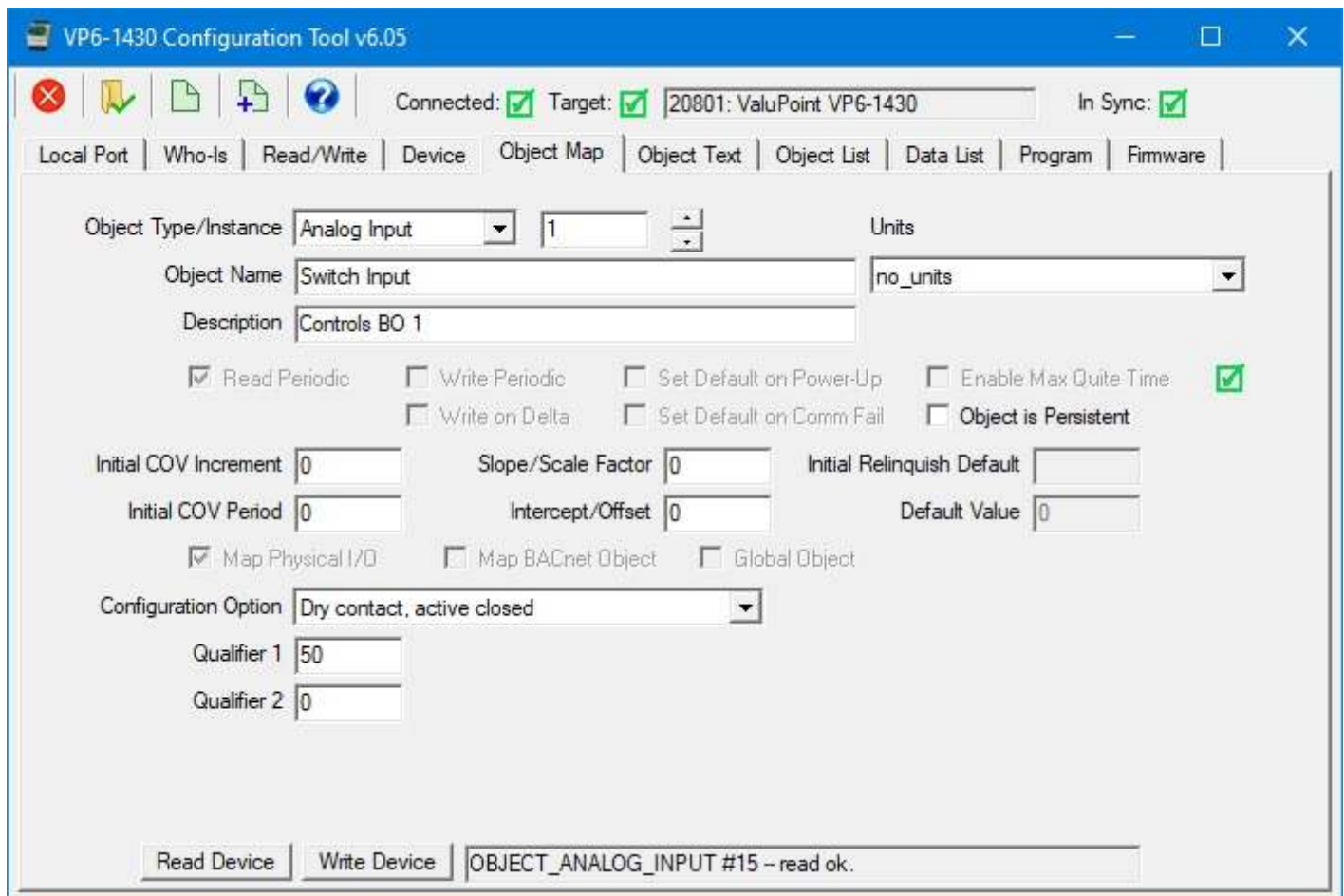


Click the green "i" icon next to the folder icon to open the i.CanDrawIt graphical programming tool that is illustrated below. It has its own set of help pages. Click on the "?" icon in that tool for more about programming. (Note that i.CanDrawIt is a second software package installed after the ValuPoint configuration tool - if you have trouble starting up i.CanDrawIt, be sure it was installed.)

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.

The above simple example program shows operating the relay output BO #1 (relay output 1) from input AI #1 (hardware input A/UI #1). To configure A/UI #1 for use in the above example, go to the Objects page and configure A/UI #1 for dry contact (or other applicable discrete type).



### 11.3 Program Capacity

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

Maximum RAM available for program variable and stack space: 16,384 bytes

Maximum EEPROM available: 256 bytes

### 11.4 Program States and Error Codes

Program State codes:

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

Reason for Halt codes (Indicated as Error Code):

- 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

## 11.5 Object and Special Register Access

BACnet objects are referenced in i.CanDrawIt programs as a calculated register number. Register numbers are BACnet object type multiplied times 1,000 plus object number starting at #1. Register numbers corresponding to BACnet objects are as follows:

Object Type	Object Number	Register Number
Analog Input	AI 1	1
Analog Output	AO 1	1001
Analog Value	AV 1	2001
Binary Input	BI 1	3001
Binary Output	BO 1	4001
Binary Value	BV 1	5001

Multi-State Input	MI 1	13001
Multi-State Output	MO 1	14001
Multi-State Value	MV 1	19001

Register number =  $n * 1000 + m$  where  $n$  is object type,  $m$  is instance. Positive register number reads/writes present value. Negative register number reads object reliability (cannot write), integer only. Object instance 0 writes the commandable priority level that will be used in subsequent writes of present value to commandable objects. (e.g. writing 8 to register 1000 will set command priority to 8 for all output objects)

Proprietary registers are  $128000 + m$  where  $m$  is special register number. Negative numbers are not allowed. Unless noted otherwise, special registers are integer only; floating point will return 0 or be discarded. Integer values will be treated as unsigned.

Special registers for reading are as follows:

- 1 = RTC Year
- 2 = RTC Month
- 3 = RTC Day of month
- 4 = RTC Day of week, 1=Mon, 7=Sun, 255=invalid
- 5 = RTC Hour
- 6 = RTC Minute
- 7 = RTC Second

Special registers for writing are as follows:

- 101 = Set object number out of service\*
- 102 = Set object number in service\*
- 103 = Reset poll timer for object to zero\* (force immediate read via client)

\* The 'write' value must be integer, and should be the register number to be affected using the register number encoding normally used to read/write present value. (e.g. write value 1001 to register 128101 to set AO 1 out of service, e.g. write value 2001 to register 128103 to reset poll timer for AV 1)



## 12 Firmware Update Page

Firmware update support is included as a page in the VP6-1430 configuration tool. The firmware update is done via BACnet MS/TP.

### 12.1 Firmware Organization

There are two copies of the application firmware - a primary and a backup image. The primary image is the one that is always running when the VP6-1430 is booted up. In the unlikely event that the primary image becomes corrupted, the backup image will replace it.

There are redundant copies of something known as a "boot information record" or "boot info" for short. There are two copies of the boot info for the primary image, and two copies of the boot info for the backup image. The primary and backup images are stored in the processor's flash memory. The boot info records are stored in on-chip EEPROM (non-volatile memory).

The boot info record contains the following information: Application code size, application image checksum, model number and variant, and firmware revision.

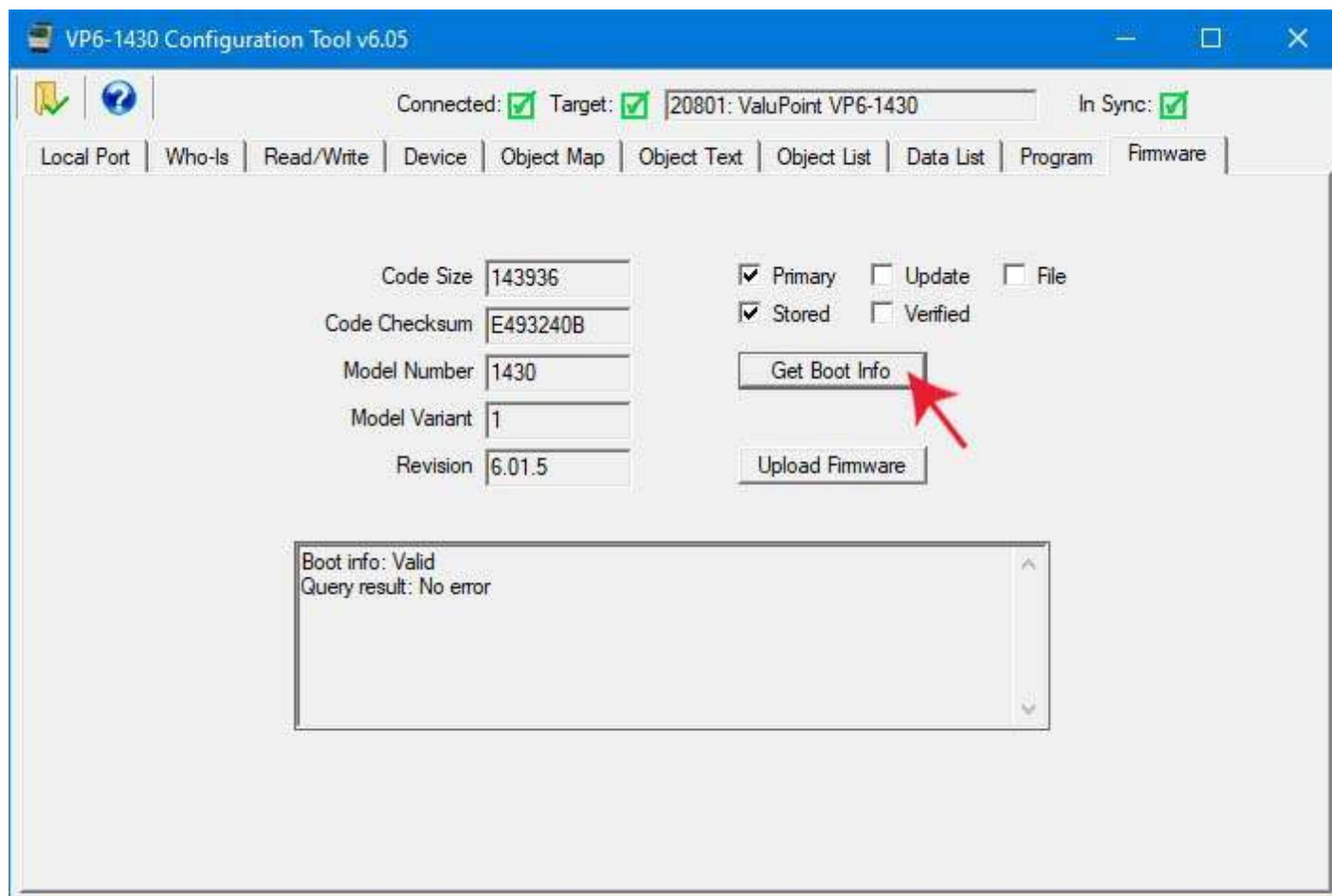
Upon bootup, the primary application image will be scanned and its checksum calculated. If this does not match the previously stored boot info record, it will be assumed that the application has somehow gotten corrupted. If the primary image is deemed to be corrupt (checksum does not match), then the backup image is scanned. If the backup image is deemed to be good, then the primary application area will be reprogrammed with a copy of the backup image.

It is generally highly unlikely that the application image flash area will become corrupted because the only time any write operations will be taking place with flash is in reprogramming firmware. If power is lost in the middle of a firmware update, for example, then the flash image will likely be corrupted.

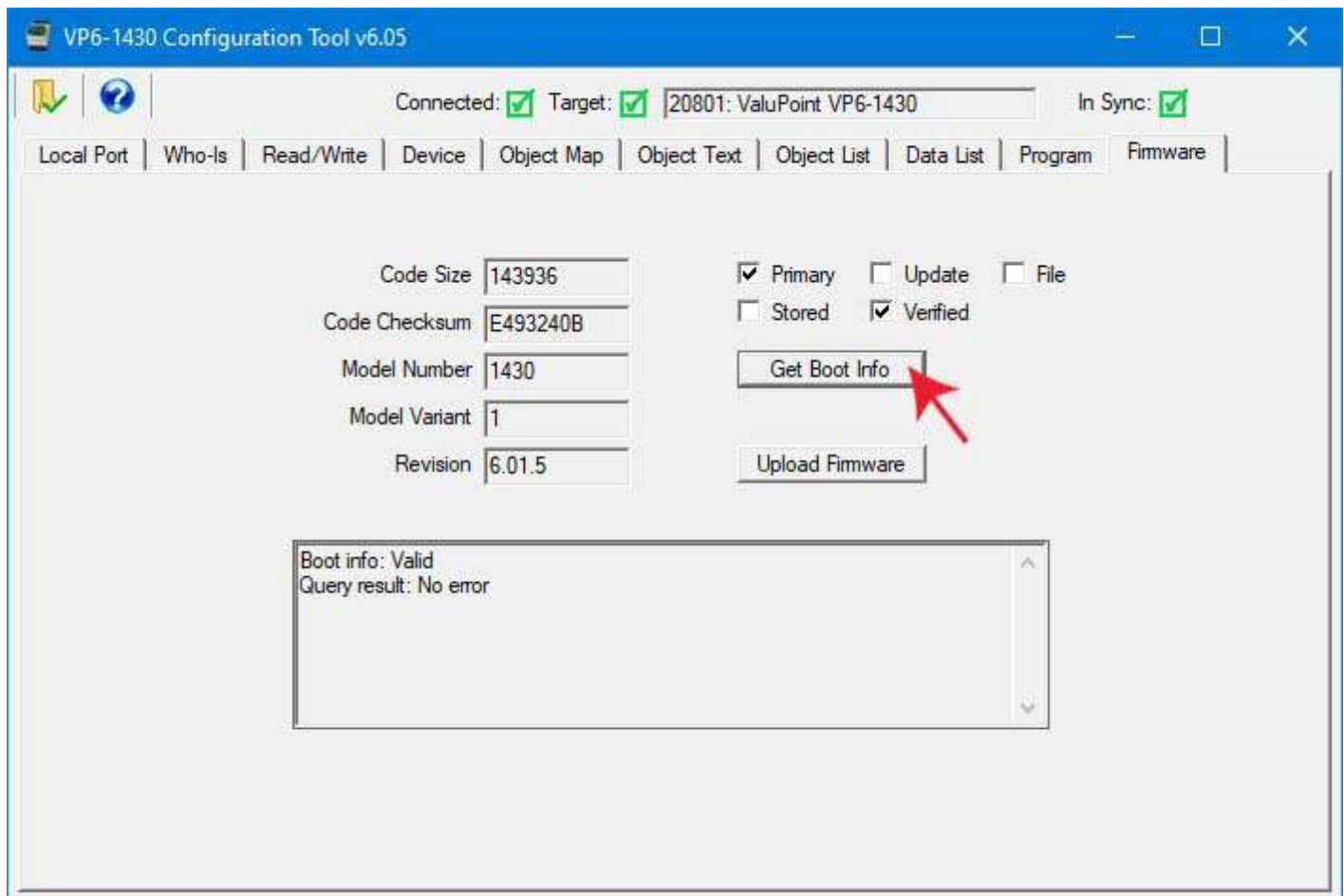
It is probably more likely that the EEPROM that stores the boot info may become corrupted by a loss of power while writing to EEPROM. Writes to EEPROM are somewhat more frequent as certain parts of the device's configuration are stored there. In the event both primary and backup images fail the checksum test, then a minimal validity check on the primary application is made. If it seems that there is a reasonable chance that the application may be valid, the boot loader will go ahead and start up the primary application, but with an error indication to alert you to the fact that this has happened.

## 12.2 Verifying Primary Application Image

If you check "Primary" and "Stored" and then click "Get Boot Info", you will retrieve the boot info record for the primary application image.

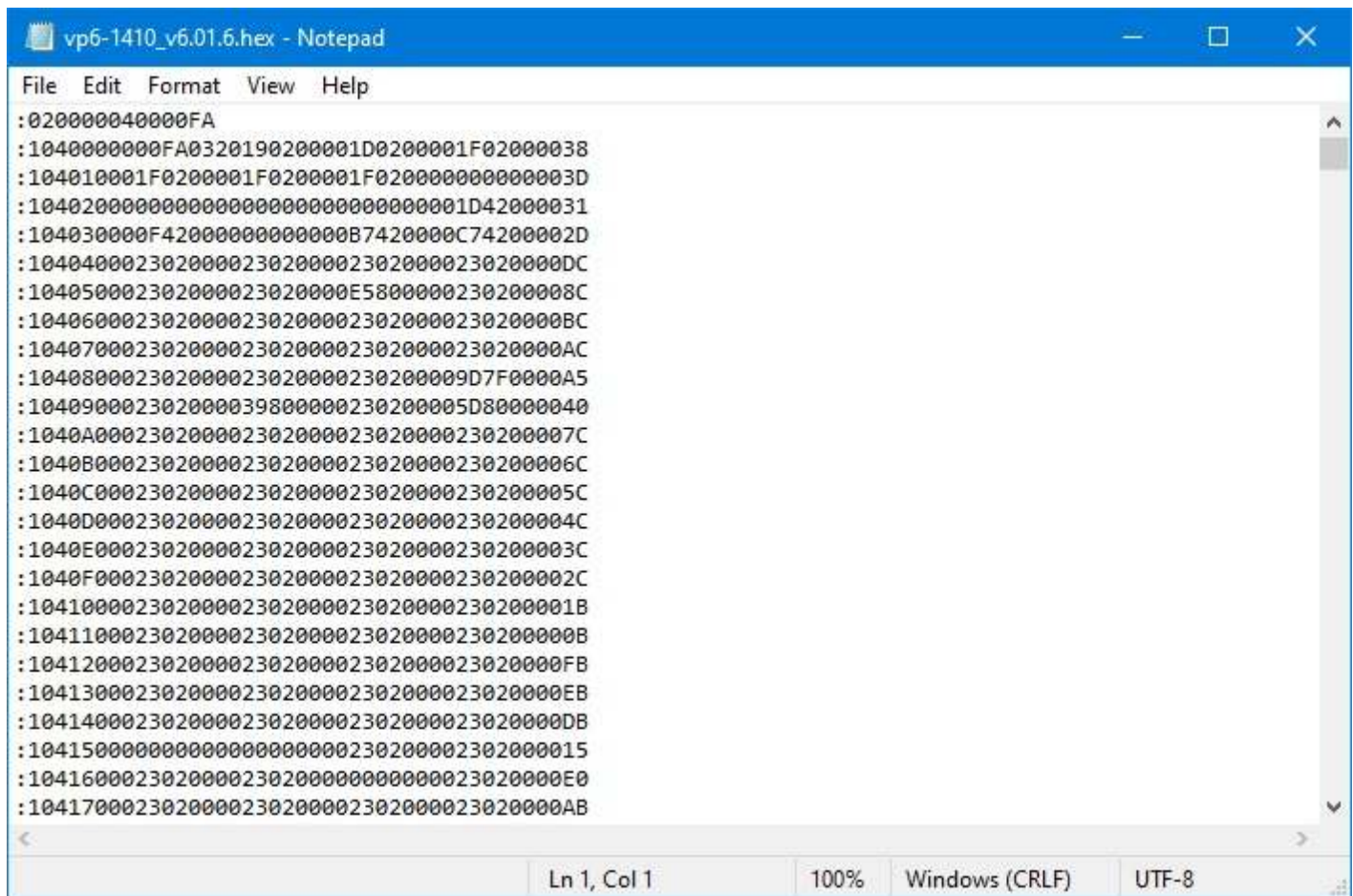


If you click "Primary" and "Verified" and then click "Get Boot Info", you will cause the device to scan its primary application image and compare it to the stored boot info record. It should always come back as valid, no error, and display the current firmware revision.



### 12.3 Firmware Update Process

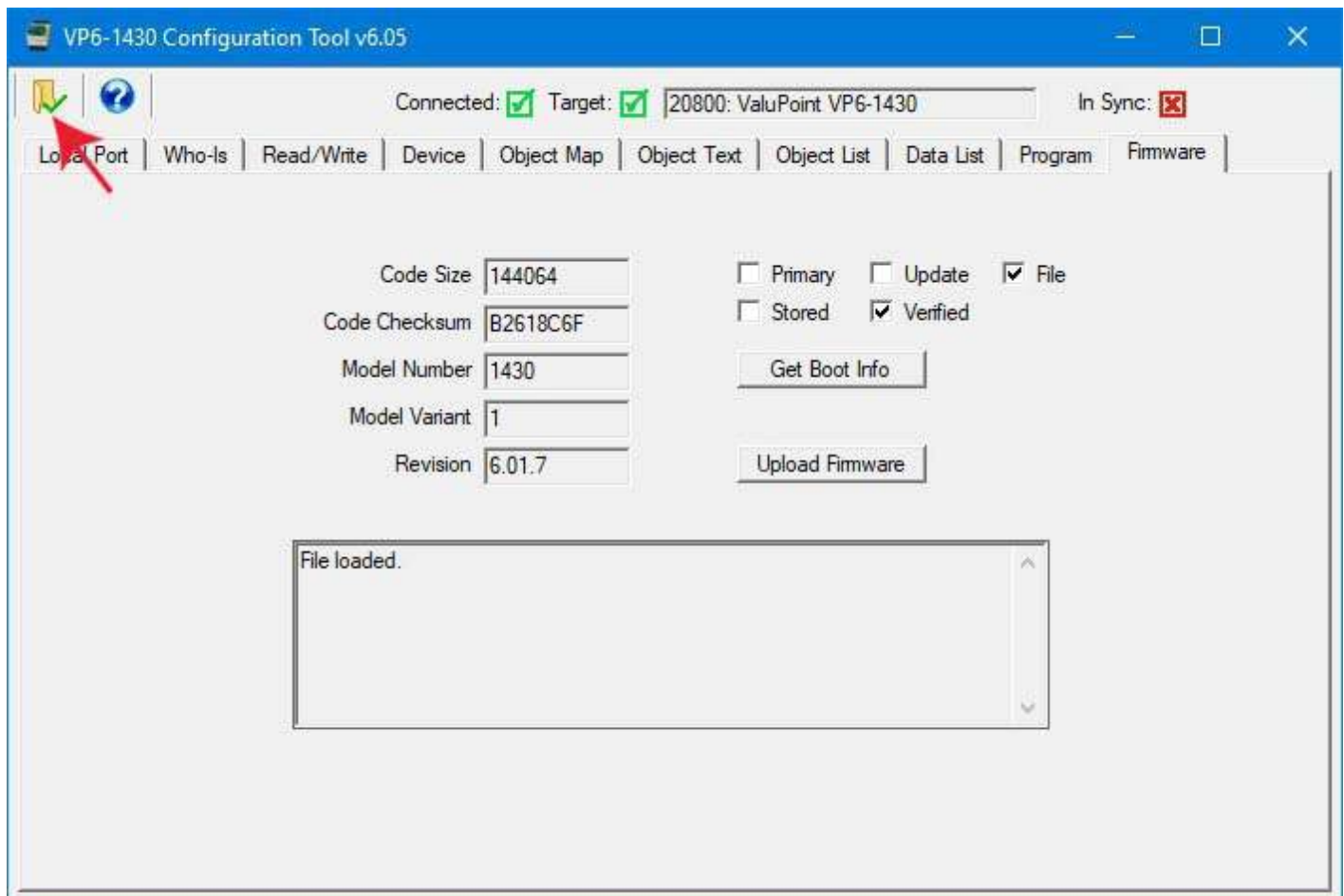
If there is newer firmware and you have good reason to believe a firmware update would resolve some issue you're having, then start by obtaining a copy of the firmware update from Control Solutions support. The update will be in the form of a "hex" file, typically named something like "VP6-1430.hex" and its contents, which will always be printable, will look something like this (abbreviated view):



```
vp6-1410_v6.01.6.hex - Notepad
File Edit Format View Help
:020000040000FA
:1040000000FA0320190200001D0200001F02000038
:104010001F0200001F0200001F02000000000003D
:104020000000000000000000000000000001D42000031
:104030000F42000000000000B7420000C74200002D
:1040400023020000230200002302000023020000DC
:104050002302000023020000E5800000230200008C
:1040600023020000230200002302000023020000BC
:1040700023020000230200002302000023020000AC
:104080002302000023020000230200009D7F0000A5
:104090002302000039800000230200005D80000040
:1040A000230200002302000023020000230200007C
:1040B000230200002302000023020000230200006C
:1040C000230200002302000023020000230200005C
:1040D000230200002302000023020000230200004C
:1040E000230200002302000023020000230200003C
:1040F000230200002302000023020000230200002C
:10410000230200002302000023020000230200001B
:10411000230200002302000023020000230200000B
:1041200023020000230200002302000023020000FB
:1041300023020000230200002302000023020000EB
:1041400023020000230200002302000023020000DB
:10415000000000000000000000230200002302000015
:1041600023020000230200000000000023020000E0
:1041700023020000230200002302000023020000AB
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

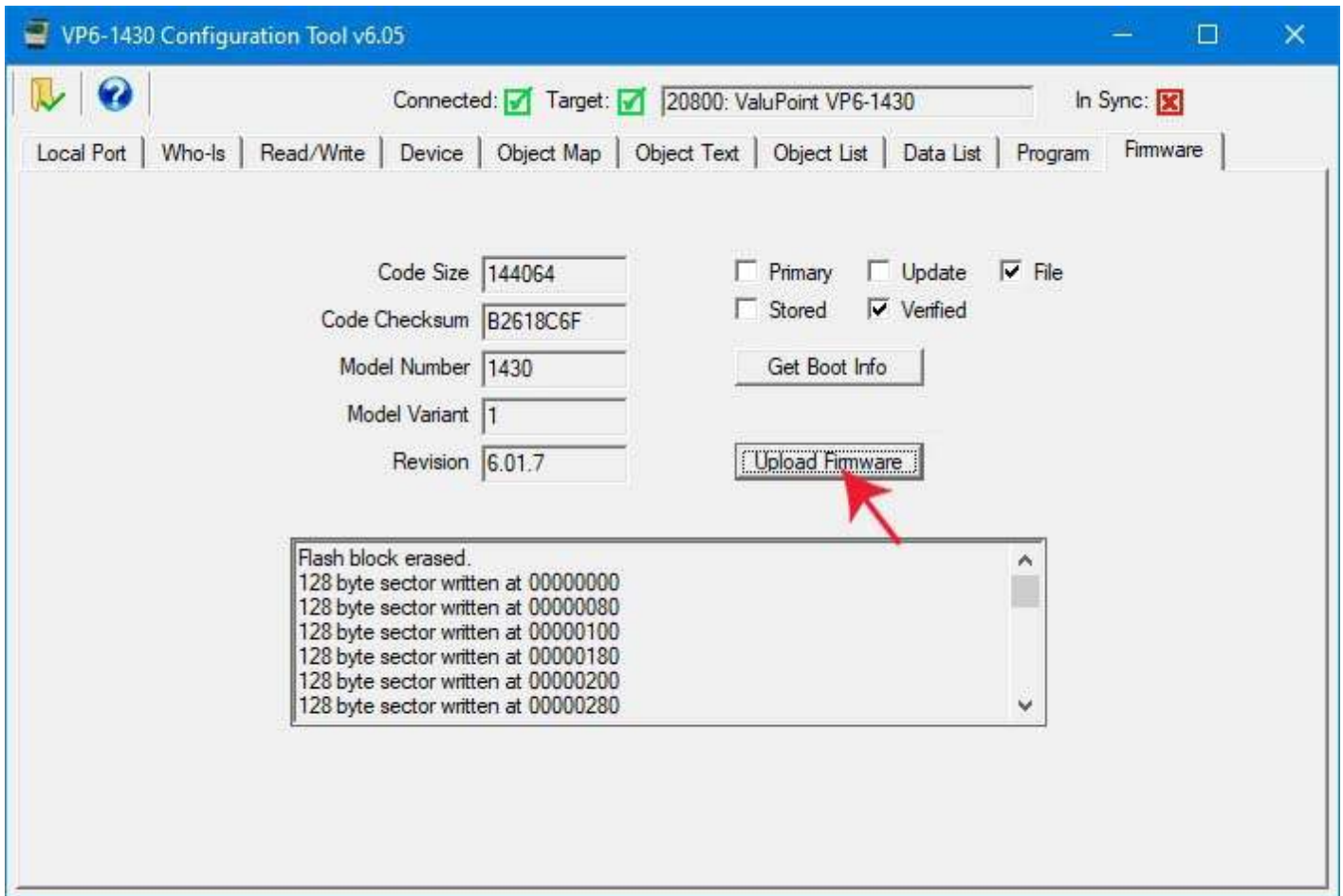
The VP6-1430 configuration tool should be connected in the same manner as for configuring I/O objects.

To upload a firmware update, start by clicking on the file open icon. Use the standard Windows file dialog that opens automatically to locate your copy of the .hex file and open it. After loading the file, the information displayed now is taken from the file just loaded. The File checkbox indicates that you are looking at boot info for the loaded code file rather than the connected device.



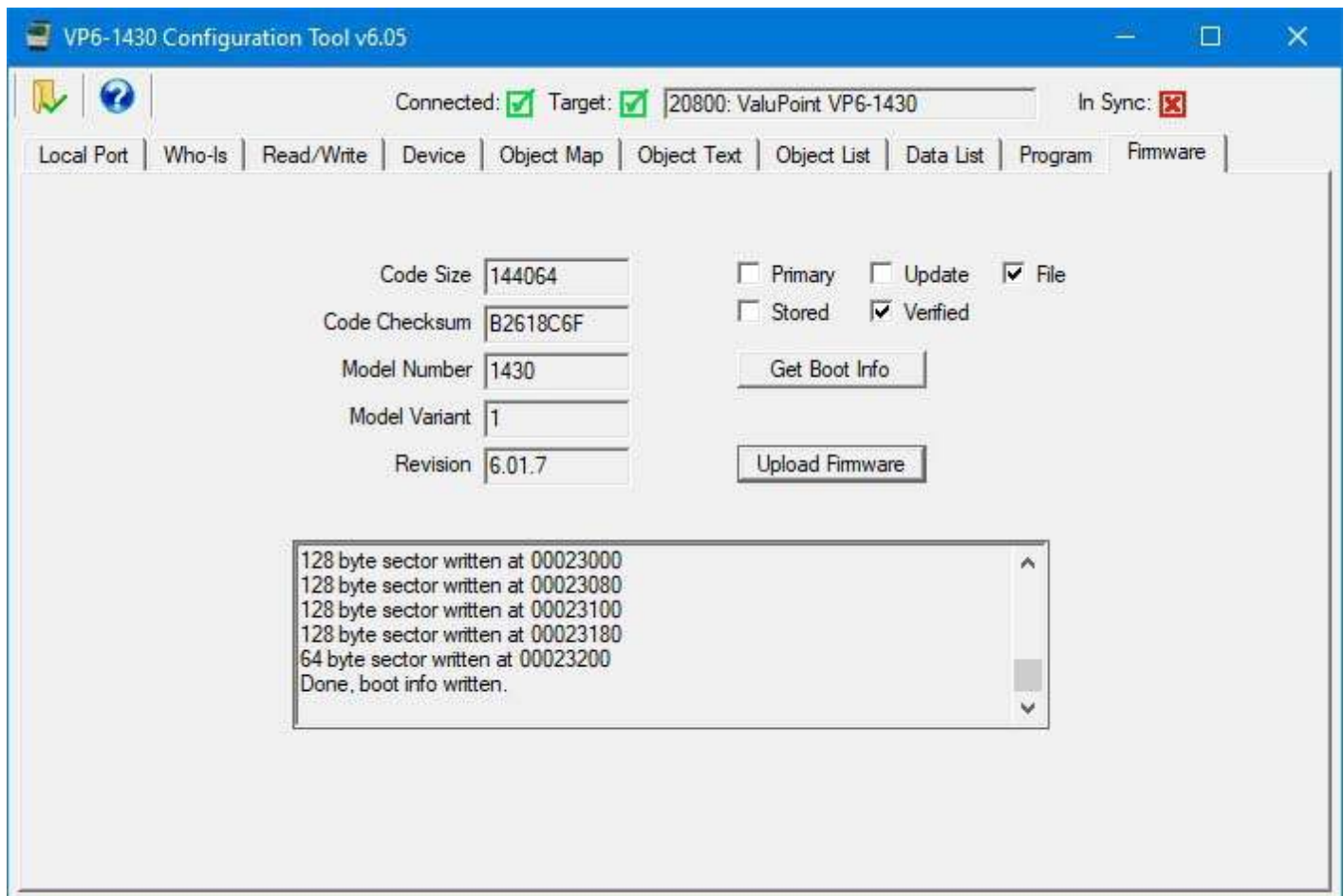
The upload process will take a couple of minutes. During this time, progress will continue to get posted to the status window. When the upload is complete, the boot info record for the update image is written.

The firmware update is programmed into the backup application area of flash memory in the processor chip. Once the upload is complete, and verified, then all you need to do is restart the device to complete the process.

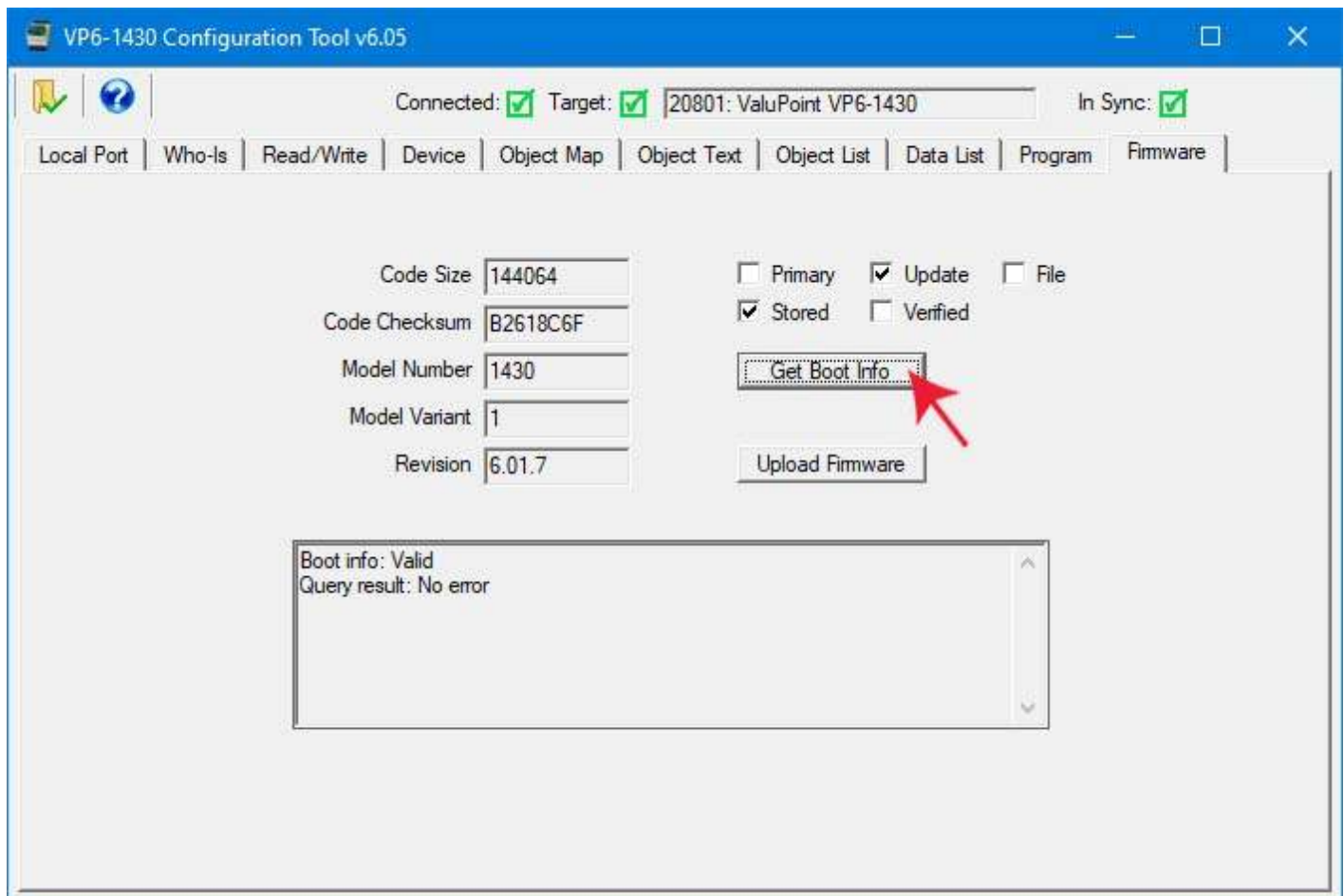


Once the upload process is complete, the "Done" message will appear.





Following the upload, you can query the stored boot info record for the update image.



By selecting "Verified", you can cause the device to scan the code just uploaded and compare the checksum. You should get "Valid, No error" and this will indicate that the upload was successful and code has been verified.

Next, you will need to restart the device. Do this by simply power cycling the VP6-1430. The device will now restart, and upon bootup, and upon seeing that a newer image is available, the backup image will be copied into the primary application flash area of memory.

Following the restart, you should see the new firmware version displayed when you select Primary and Stored and then click the Get Boot Info button.



## Appendix A - Hardware Details

### A.1 Connection of Inputs

The VP6-1430 contains no configuration jumpers for configuring I/O points. There is no need to open the enclosure for configuration of I/O. Input types are switched 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 Objects page of the configuration tool.

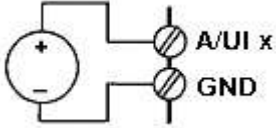
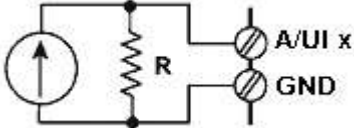
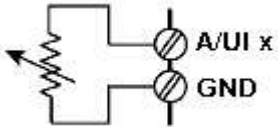
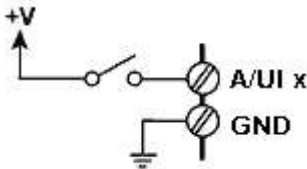
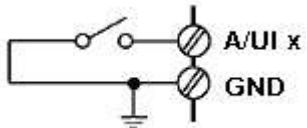
I/O Point Type	Wiring Guide	Additional Information
<b>A/UI Analog Input: 0-10VDC Voltage Input or Pulse Count</b>		<p>A/UI inputs 1-12 will accept voltage inputs of up to 10VDC. Voltages to 11VDC will be measured. Voltages to 24VDC will be tolerated, but measurement is internally limited to a reading of 11VDC.</p> <p>This wiring diagram is also applicable to pulse counter input for counting pulses from an active pulse generator.</p>
<b>A/ UI Analog Input: 0-20mA Current Input</b>		<p>A/UI inputs 1-12 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.</p>
<b>A/UI Analog Input: Thermistor Input</b>		<p>A/UI inputs 1-12 will accept thermistors of 3k, 5k, 10k, or 20k ohms. Linearization via interpolation of a 56-point</p>

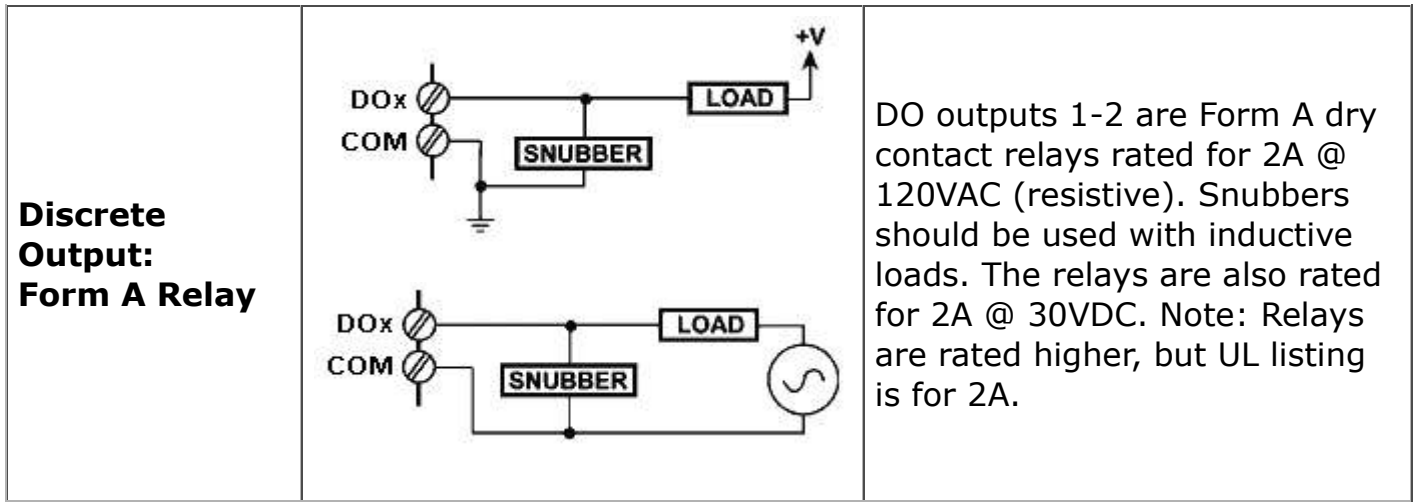
		table is performed internally.
<b>A/UI or DI Discrete Input: Discrete Voltage</b>		A/UI inputs 1-12 may be connected as discrete voltage sensing inputs. Inputs up to 24VDC are tolerated, but threshold sensing only functions over the 0-10VDC range.
<b>A/UI or DI Discrete Input: Dry Contact Closure to Ground or Pulse Count</b>		<p>A/UI inputs 1-12 may be connected as discrete inputs sensing dry contact closure to ground. Internal excitation of 0.3mA is provided. The excitation current may be increased by the addition of an external resistor (pullup to DC power).</p> <p>This wiring diagram is also applicable to pulse counter input for counting pulses from a switch closure.</p>

## A.2 Connection of Outputs

The VP6-1430 contains no configuration jumpers for configuring outputs. There is no need to open the enclosure.

Output points should be connected as indicated in the diagrams below. The discrete (relay) outputs are SPST N.O. with the common side connected together to the COM terminal. The COM terminal for relays is NOT electrically common to any of the GND terminals.

I/O Point Type	Wiring Guide	Additional Information
----------------	--------------	------------------------

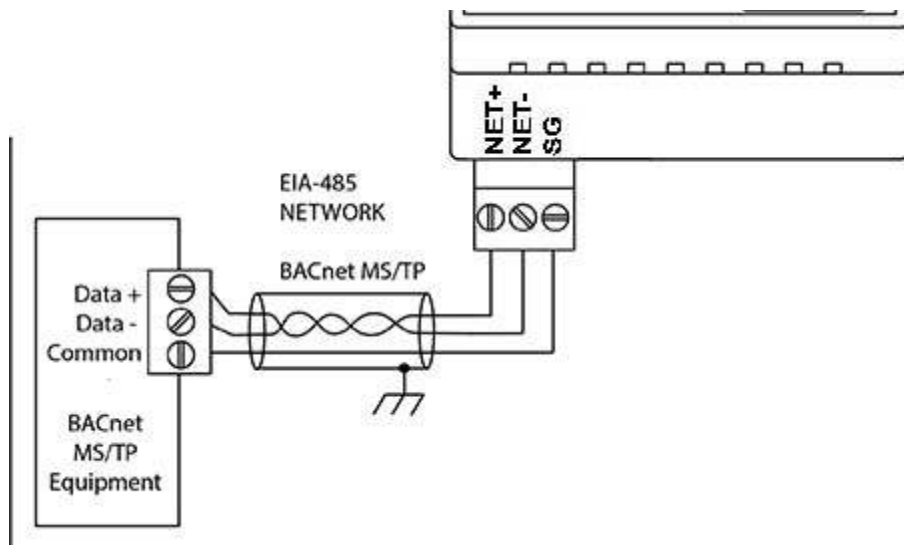


### A.3 Connection of Power and Communications

Power and communications should be connected as indicated below.

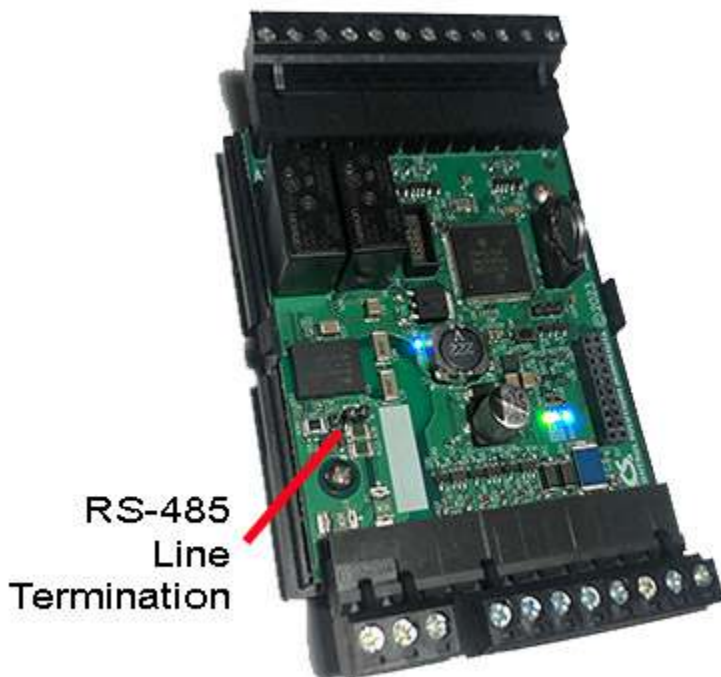
I/O Point Type	Wiring Guide	Additional Information																																																
<p><b>Power</b></p>	<p>Connect AC or +DC power to Power terminal. Connect common or -DC power to GND terminal. GND is common to all terminals labeled COM.</p>	<p>Nominal power consumption is 2.4 watts, or 0.1A @ 24VDC, with all relays on.</p>																																																
<p><b>Communications</b></p>	<p>Connect BACnet MS/TP RS-485 data lines to NET+/- terminals. The RS-485 port is electrically isolated from all other terminals including power. Therefore, the SG must be connected to the ground reference for the RS-485 signal in order to communicate.</p>	<p>Communication signals comply with EIA-485 standard.</p>																																																
<p><b>Wiring Terminals</b></p>	<table border="0" style="width: 100%; text-align: center;"> <tr> <td>RS-485</td> <td>NET+</td> <td>COM</td> <td>Relay</td> </tr> <tr> <td>BACnet MS/TP</td> <td>NET-</td> <td>DO 1</td> <td>Outputs</td> </tr> <tr> <td></td> <td>SG</td> <td>DO 2</td> <td></td> </tr> <tr> <td></td> <td></td> <td>GND</td> <td></td> </tr> <tr> <td></td> <td>A/UI 1</td> <td>A/UI 7</td> <td></td> </tr> <tr> <td></td> <td>A/UI 2</td> <td>A/UI 8</td> <td></td> </tr> <tr> <td>Sensor</td> <td>A/UI 3</td> <td>A/UI 9</td> <td>Sensor</td> </tr> <tr> <td>Inputs</td> <td>A/UI 4</td> <td>GND</td> <td>Inputs</td> </tr> <tr> <td></td> <td>A/UI 5</td> <td>A/UI 10</td> <td></td> </tr> <tr> <td></td> <td>A/UI 6</td> <td>A/UI 11</td> <td></td> </tr> <tr> <td>Power</td> <td>POWER</td> <td>A/UI 12</td> <td></td> </tr> <tr> <td>Input</td> <td>GND</td> <td>GND</td> <td></td> </tr> </table>	RS-485	NET+	COM	Relay	BACnet MS/TP	NET-	DO 1	Outputs		SG	DO 2				GND			A/UI 1	A/UI 7			A/UI 2	A/UI 8		Sensor	A/UI 3	A/UI 9	Sensor	Inputs	A/UI 4	GND	Inputs		A/UI 5	A/UI 10			A/UI 6	A/UI 11		Power	POWER	A/UI 12		Input	GND	GND		<p>Screw terminals ratings are substantially in excess of any I/O point ratings.</p> <p>Screw terminals are pluggable. They unplug from the unit in 3, 8, and 12-position blocks.</p>
RS-485	NET+	COM	Relay																																															
BACnet MS/TP	NET-	DO 1	Outputs																																															
	SG	DO 2																																																
		GND																																																
	A/UI 1	A/UI 7																																																
	A/UI 2	A/UI 8																																																
Sensor	A/UI 3	A/UI 9	Sensor																																															
Inputs	A/UI 4	GND	Inputs																																															
	A/UI 5	A/UI 10																																																
	A/UI 6	A/UI 11																																																
Power	POWER	A/UI 12																																																
Input	GND	GND																																																

The RS-485 should be wired as illustrated below.



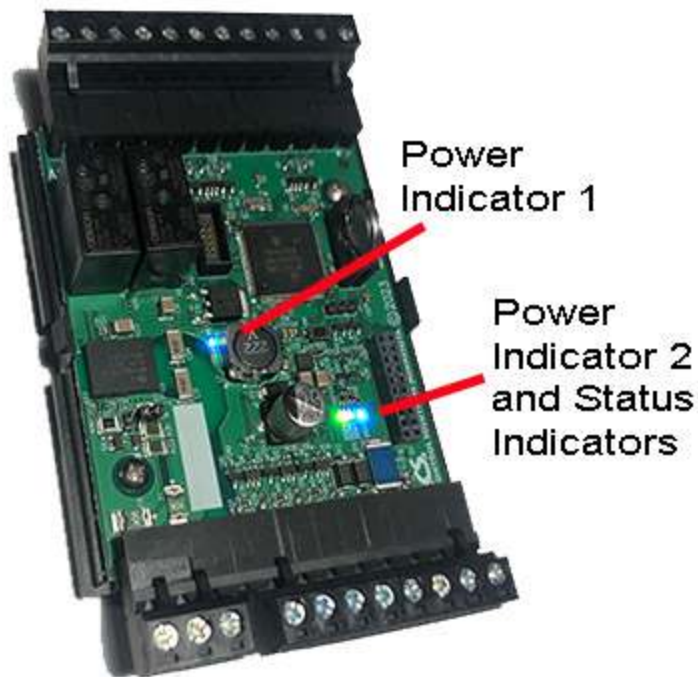
#### A.4 RS-485 Line Termination

RS-485 line termination jumper is located as indicated in the photo. The line termination is "on" when the jumper block is aligned with the screened white block on the circuit board. Termination is required when the VP6-1430 is the last device on the daisy chain.



#### A.5 LED Indicators

LED indicators are depicted in the following photo. Status indicators include green, red, and yellow LEDs which are visible through the vent slots of the enclosure. Two blue power indicators are also visible through the vent slots.



Status indicator functions for BACnet MS/TP:

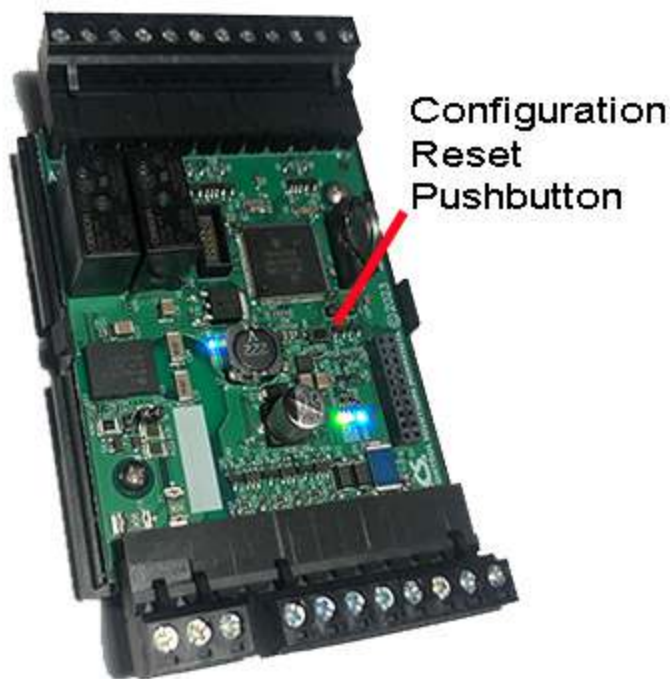
Green LED	Yellow LED	Red LED
Flashes upon token pass or packet send/receive	Flashes upon Poll for Master sent by this device	Flashes upon error in packet

NOTE: If only the red LED is flashing, and is primarily on while flashing briefly off a few times, count the number of "off" flashes. This is a system error code. Refer to the trouble shooting section.

## A.6 Restore Factory Defaults

The configuration reset pushbutton is located on the circuit board where indicated below.

**IMPORTANT:** Using the reset button should be your last resort. If you have simply forgotten what baud rate or address the VP6-1430 is set to, simply use Auto Connect as outlined in section 3 of this user guide. Clearing all configuration should not really be required since you overwrite all old configuration every time you write a new configuration using the configuration tool.



There are two "codes" for restoring factory default.

Code 1: Reset communication parameters, but leave configuration intact.

Code 2: Reset communication parameters and erase all configuration.

Press and hold the button until you see code 1 flashing. If you wish to perform code 1, release the button while code 1 is flashing. Otherwise continue to hold the button until you see code 2 flashing and then release the button. If you do not release the button while code 2 is flashing, it will return to code 1 in a few seconds, and the codes will continue to alternate for as long as you hold the button down.

Upon releasing the button, you will see the code confirmation flashing. Press the button again to cancel the action. You must do the cancel within the next few seconds, otherwise the action will be performed. If you do not cancel the action, the action will be performed in a few seconds.

The LEDs that normally show MS/TP traffic are temporarily used to display these codes while the reset is pending.

Code 1: Yellow LED flashes once and then pause (and then repeat).

Code 2: Red LED flashes twice and then pause (and then repeat).

Confirmation: Green LED flashes showing either code 1 or code 2 (one or two flashes followed by a pause, and then repeat).

## **A.7 Battery Replacement**

No action is required of the user to activate the battery that backs up the real time



clock. The battery should have a 10-year life in normal operation with "normal" meaning that the device is normally continuously powered for use with only intermittent dependency on the battery backup.



Replace battery with BR1225A only. Use of another battery may present a risk of fire or explosion. Replace with battery polarity as marked on the circuit board. Reverse polarity of the battery will not damage the board, but the clock will not be backed up if the battery is reversed.

**CAUTION:** The lithium battery contained in this device may explode if mistreated. DO NOT recharge, disassemble, or dispose of in fire.



## Appendix B - BACnet Object Properties

### B.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> configuration property fault (80) 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)
Active_Text (4)	CharacterString (binary objects only)
Inactive_Text (46)	CharacterString (binary objects only)
State_Text (110)	CharacterString (multistate 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) object value is persistent (3) object is reserved for global use (4) set default on power-up (5) set default on comm. fail (6) read periodic (7) write periodic (8) write on delta (9) enable max quiet
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_COV_Period (817) (W)	Unsigned
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_MAC (827)	Slave Mac address if who-is not supported (0 means unused, cannot use Mac=0)
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

## B.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>	
MSTP_Port_Baud_Rate (1201) (W)	<p>ENUMERATED: (9600, 19200, 38400, 76800) NOTE: Changes to port settings will take effect only after a COLDSTART or WARMSTART command is issued.</p>
MSTP_Station_ID (1202) (W)	Unsigned
Enable_Auto_Reset (1203) (W)	<p>BOOLEAN True if reliability codes should auto return to zero</p>
Alt_Map (1204) (W)	<p>BOOLEAN Remaps relay outputs DO1 and DO2 to Binary Value objects instead of Binary Output objects if true</p>
Reinit_Password (1205) (WO)	<p>CharacterString Note: This property is write-only, and is only writeable with unlock sequence.</p>
Password_Unlock (1206) (W)	<p>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.</p>
Device_Allocation (1251) (W)	<p>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)</p>

(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.***





## Appendix C - CSV File Format

### C.1 Object Maps

The columns noted in the following table are used to identify and define BACnet objects in the gateway device.

Object	<p>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:</p> <ul style="list-style-type: none"> <li>AI = Analog Input</li> <li>AO = Analog Output</li> <li>AV = Analog Value</li> <li>BI = Binary Input</li> <li>BO = Binary Output</li> <li>BV = Binary Value</li> <li>MI = Multi-state Input</li> <li>MO = Multi-state Output</li> <li>MV = Multi-state Value</li> </ul> <p>Example: "AI-1" is Analog Input 1, "AO-14" is Analog Output 14, etc. (You do not need to include quote characters for this column in the CSV file.)</p>
IsBACnet	T if mapped to BACnet objects in another BACnet device, or F
ReadPoll	T if periodic Read, or F
WritePoll	T if periodic Write, or F
WriteDelta	T if Write on Delta, or F
DefPOR	T if default on POR enabled, or F
DefNOK	T if default on comm. fail enabled, or F
PollTime	Integer, Periodic poll time in seconds
Timeout	Integer, BACnet slave timeout in seconds
MaxQuiet	T if max. quiet enabled, or F
Scale	Real, scale factor
Offset	Real, offset for scaling

DefaultValue	Real or Integer as applicable to object type, default value
SendDelta	Real, delta threshold for Write on Delta
InitCOVincrement	Real, initial COV increment
InitCOVperiod	Integer, COV period in seconds
InitRelinquish	Real or Integer as applicable to object type, initial relinquish default
FailCount	Integer, count of comm. fails before Fault indicated
Units	Integer, Units code as defined by BACnet standard
ObjectName	Character String (must enclose in quotes if string includes comma character)
Description	Character String (must enclose in quotes if string includes comma character)

The columns noted in the following table are used to identify BACnet objects in other BACnet devices that will be polled by the gateway device, with content of the polled objects placed into (or taken from) BACnet objects in the gateway device.

ObjDevice	Integer, BACnet device instance
ObjType	ASCII string (2 characters) identifying the local object type, where string 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
ObjInstance	Integer, BACnet object instance
ObjProperty	Integer, BACnet property code as defined by BACnet standard
ObjIndex	Integer, array index, -1 if no index
ObjPriority	Integer, priority 1-16 or 0 if none
ObjEncoding	Integer, object encoding (0) Null (1) Boolean (2) Unsigned Integer (3) Signed Integer (4) Real

	(8) Bit String (9) Enumerated
BitNumber	Integer, bit number
MacAddr	Integer (0..254), MAC address of MS/TP device if slave-only device or does not support dynamic binding (Who-Is/I-Am). Normally should be zero so that Who-Is is allowed to take care of device discovery.

## C.2 Object Text

The columns noted in the following table are used to identify and define text strings for Binary and Multi-state objects in the gateway device.

Object	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: BI = Binary Input BO = Binary Output BV = Binary Value MI = Multi-state Input MO = Multi-state Output MV = Multi-state Value Example: "BI-1" is Binary Input 1, "BO-14" is Binary Output 14, etc. (You do not need to include quote characters for this column in the CSV file.)
State	For Binary objects: 0=inactive text, 1=active text For Multi-state objects: 1..N where N is maximum state configured for Multi-state objects
Text	Character String (must enclose in quotes if string includes comma character)



## Appendix D - BACnet Trouble Shooting

### D.1 Observing BACnet Errors, LED Indicators

The reliability code displayed for each object on the Data List page (see section 10) will indicate BACnet client errors on a point by point basis. The reliability code generally indicates the nature of the problem.

Refer to Appendix A, Hardware Details, for a detailed description of what the LEDs are indicating.

### D.2 Error Codes, Reliability Codes

If the remote BACnet device being queried by the ValuPoint returns an error code, or there is simply a problem in communicating with the remote device, this will be indicated by the ValuPoint object's reliability code. When reliability code is non-zero, the fault flag is also set. Therefore, the Status Flags indication will typically be "F,T,F,F" any time the reliability code is something other than zero (zero means no errors to report).

Error codes returned by a BACnet slave device are encoded into reliability codes presented by the respective object. Reliability codes pertaining to polling of BACnet slaves (read via the Reliability property) are as follows:

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). Refer to Appendix E.3 for a listing of BACnet error classes and codes.

### D.3 Auto-Reset Errors

Reliability codes will "latch" by default and require that you read the Reliability property in order to reset it to zero, assuming the problem has gone away. Once the non-zero reliability code has been read (by reading the Reliability property), it will reset to zero the next time the object is updated, provided the problem has been resolved.

Since many systems do not automatically read Reliability codes, but do automatically

respond to the Fault Status Flag associated with the non-zero reliability code, an auto-reset option is available (on the Device page). 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.

Many systems will report an object as "offline" because its fault status bit is set. It is not actually offline, and is in fact communicating just fine trying to tell you that there is a problem. But many front end systems don't recognize this and blindly claim "offline" as a result of the fault bit in the status property. If you are having this issue as the result of communication errors related to polling other BACnet devices, try setting the Auto-Reset Errors option here. This only applies if you have "Map BACnet Object" checked on one or more Object Map pages.

#### D.4 Trouble Shooting Tips, 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.

**TROUBLE SHOOTING TIP:** If LEDs are not flashing on the MTX002 and/or you cannot discover any device on the Who-Is page, try disconnecting the MTX002 and restarting the configuration tool. Then, instead of selecting a baud rate and clicking Connect, just click Auto Connect. This will search for MS/TP on all possible baud rates, automatically determine the network's Max Master setting, and automatically assign an unused MAC address to the MTX002. When the MTX002 has found MS/TP (after a minute or two), it will say "Auto connected" where it normally just says "Ready". Discovering that the ValuPoint is not set for the baud rate somebody thought it was is a common technical support finding.

## **D.5 Trouble Shooting Tips, ValuPoint 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 ValuPoint 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 VP6-1430 MS/TP Connection

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

(b) Are any yellow or green LEDs flickering inside the VP6-1430? The yellow and/or green LEDs inside the VP6-1430 should be flickering 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 VP6-1430 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.

## **D.6 System Fault Indications**

The red LED visible inside the VP6-1430 case, viewed through the vent slots, near the blue power LED, doubles as the system fault indicator. It will be on during initial power-up boot mode operation, but should otherwise be off.

Following a device reinitialize upon command from BACnet, the system fault LED will flicker at a very rapid rate for approximately 10 seconds, then turn off (or indicate system fault). The rapid flicker is verification that the restart has occurred.

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) Processor hard fault
- (2) Unhandled IRQ
- (3) Stack overflow
- (4) Other OS fault
- (5) Resource allocation fail
- (6) System configuration checksum error
- (7) EEPROM read/write fail
- (8) Flash read/write fail
- (9) Application checksum error
- (10) Invalid application image
- (11) Unknown fault

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 may indicate a hardware failure requiring factory attention. You should really never see any fault codes.



## Appendix E - BACnet Codes

### E.1 BACnet Object Property Codes

BACnet property type codes may be found in your copy of the BACnet protocol specification, ANSI/ASHRAE Standard 135. That document is copyrighted, but the C enumeration shown below for reference is taken from open source code available under GPL at <http://sourceforge.net>, and provides essentially the same information (copyrighted by Steve Karg, licensed under GPL as noted at <http://sourceforge.net>).

```
typedef enum {
    PROP_ACKED_TRANSITIONS = 0,
    PROP_ACK_REQUIRED = 1,
    PROP_ACTION = 2,
    PROP_ACTION_TEXT = 3,
    PROP_ACTIVE_TEXT = 4,
    PROP_ACTIVE_VT_SESSIONS = 5,
    PROP_ALARM_VALUE = 6,
    PROP_ALARM_VALUES = 7,
    PROP_ALL = 8,
    PROP_ALL_WRITES_SUCCESSFUL = 9,
    PROP_APDU_SEGMENT_TIMEOUT = 10,
    PROP_APDU_TIMEOUT = 11,
    PROP_APPLICATION_SOFTWARE_VERSION = 12,
    PROP_ARCHIVE = 13,
    PROP_BIAS = 14,
    PROP_CHANGE_OF_STATE_COUNT = 15,
    PROP_CHANGE_OF_STATE_TIME = 16,
    PROP_NOTIFICATION_CLASS = 17,
    PROP_BLANK_1 = 18,
    PROP_CONTROLLED_VARIABLE_REFERENCE = 19,
    PROP_CONTROLLED_VARIABLE_UNITS = 20,
    PROP_CONTROLLED_VARIABLE_VALUE = 21,
    PROP_COV_INCREMENT = 22,
    PROP_DATE_LIST = 23,
    PROP_DAYLIGHT_SAVINGS_STATUS = 24,
    PROP_DEADBAND = 25,
    PROP_DERIVATIVE_CONSTANT = 26,
    PROP_DERIVATIVE_CONSTANT_UNITS = 27,
    PROP_DESCRIPTION = 28,
    PROP_DESCRIPTION_OF_HALT = 29,
    PROP_DEVICE_ADDRESS_BINDING = 30,
    PROP_DEVICE_TYPE = 31,
    PROP_EFFECTIVE_PERIOD = 32,
    PROP_ELAPSED_ACTIVE_TIME = 33,
    PROP_ERROR_LIMIT = 34,
    PROP_EVENT_ENABLE = 35,
    PROP_EVENT_STATE = 36,
    PROP_EVENT_TYPE = 37,
    PROP_EXCEPTION_SCHEDULE = 38,
    PROP_FAULT_VALUES = 39,
```



```
PROP_FEEDBACK_VALUE = 40,  
PROP_FILE_ACCESS_METHOD = 41,  
PROP_FILE_SIZE = 42,  
PROP_FILE_TYPE = 43,  
PROP_FIRMWARE_REVISION = 44,  
PROP_HIGH_LIMIT = 45,  
PROP_INACTIVE_TEXT = 46,  
PROP_IN_PROCESS = 47,  
PROP_INSTANCE_OF = 48,  
PROP_INTEGRAL_CONSTANT = 49,  
PROP_INTEGRAL_CONSTANT_UNITS = 50,  
PROP_ISSUE_CONFIRMED_NOTIFICATIONS = 51,  
PROP_LIMIT_ENABLE = 52,  
PROP_LIST_OF_GROUP_MEMBERS = 53,  
PROP_LIST_OF_OBJECT_PROPERTY_REFERENCES = 54,  
PROP_LIST_OF_SESSION_KEYS = 55,  
PROP_LOCAL_DATE = 56,  
PROP_LOCAL_TIME = 57,  
PROP_LOCATION = 58,  
PROP_LOW_LIMIT = 59,  
PROP_MANIPULATED_VARIABLE_REFERENCE = 60,  
PROP_MAXIMUM_OUTPUT = 61,  
PROP_MAX_APDU_LENGTH_ACCEPTED = 62,  
PROP_MAX_INFO_FRAMES = 63,  
PROP_MAX_MASTER = 64,  
PROP_MAX_PRES_VALUE = 65,  
PROP_MINIMUM_OFF_TIME = 66,  
PROP_MINIMUM_ON_TIME = 67,  
PROP_MINIMUM_OUTPUT = 68,  
PROP_MIN_PRES_VALUE = 69,  
PROP_MODEL_NAME = 70,  
PROP_MODIFICATION_DATE = 71,  
PROP_NOTIFY_TYPE = 72,  
PROP_NUMBER_OF_APDU_RETRIES = 73,  
PROP_NUMBER_OF_STATES = 74,  
PROP_OBJECT_IDENTIFIER = 75,  
PROP_OBJECT_LIST = 76,  
PROP_OBJECT_NAME = 77,  
PROP_OBJECT_PROPERTY_REFERENCE = 78,  
PROP_OBJECT_TYPE = 79,  
PROP_OPTIONAL = 80,  
PROP_OUT_OF_SERVICE = 81,  
PROP_OUTPUT_UNITS = 82,  
PROP_EVENT_PARAMETERS = 83,  
PROP_POLARITY = 84,  
PROP_PRESENT_VALUE = 85,  
PROP_PRIORITY = 86,  
PROP_PRIORITY_ARRAY = 87,  
PROP_PRIORITY_FOR_WRITING = 88,  
PROP_PROCESS_IDENTIFIER = 89,  
PROP_PROGRAM_CHANGE = 90,  
PROP_PROGRAM_LOCATION = 91,  
PROP_PROGRAM_STATE = 92,  
PROP_PROPORTIONAL_CONSTANT = 93,  
PROP_PROPORTIONAL_CONSTANT_UNITS = 94,  
PROP_PROTOCOL_CONFORMANCE_CLASS = 95, /* deleted in version 1 revision 2 */  
PROP_PROTOCOL_OBJECT_TYPES_SUPPORTED = 96,  
PROP_PROTOCOL_SERVICES_SUPPORTED = 97,  
PROP_PROTOCOL_VERSION = 98,
```

```
PROP_READ_ONLY = 99,  
PROP_REASON_FOR_HALT = 100,  
PROP_RECIPIENT = 101,  
PROP_RECIPIENT_LIST = 102,  
PROP_RELIABILITY = 103,  
PROP_RELINQUISH_DEFAULT = 104,  
PROP_REQUIRED = 105,  
PROP_RESOLUTION = 106,  
PROP_SEGMENTATION_SUPPORTED = 107,  
PROP_SETPOINT = 108,  
PROP_SETPOINT_REFERENCE = 109,  
PROP_STATE_TEXT = 110,  
PROP_STATUS_FLAGS = 111,  
PROP_SYSTEM_STATUS = 112,  
PROP_TIME_DELAY = 113,  
PROP_TIME_OF_ACTIVE_TIME_RESET = 114,  
PROP_TIME_OF_STATE_COUNT_RESET = 115,  
PROP_TIME_SYNCHRONIZATION_RECIPIENTS = 116,  
PROP_UNITS = 117,  
PROP_UPDATE_INTERVAL = 118,  
PROP_UTC_OFFSET = 119,  
PROP_VENDOR_IDENTIFIER = 120,  
PROP_VENDOR_NAME = 121,  
PROP_VT_CLASSES_SUPPORTED = 122,  
PROP_WEEKLY_SCHEDULE = 123,  
PROP_ATTEMPTED_SAMPLES = 124,  
PROP_AVERAGE_VALUE = 125,  
PROP_BUFFER_SIZE = 126,  
PROP_CLIENT_COV_INCREMENT = 127,  
PROP_COV_RESUBSCRIPTION_INTERVAL = 128,  
PROP_CURRENT_NOTIFY_TIME = 129,  
PROP_EVENT_TIME_STAMPS = 130,  
PROP_LOG_BUFFER = 131,  
PROP_LOG_DEVICE_OBJECT = 132,  
/* The enable property is renamed from log-enable in  
Addendum b to ANSI/ASHRAE 135-2004(135b-2) */  
PROP_ENABLE = 133,  
PROP_LOG_INTERVAL = 134,  
PROP_MAXIMUM_VALUE = 135,  
PROP_MINIMUM_VALUE = 136,  
PROP_NOTIFICATION_THRESHOLD = 137,  
PROP_PREVIOUS_NOTIFY_TIME = 138,  
PROP_PROTOCOL_REVISION = 139,  
PROP_RECORDS_SINCE_NOTIFICATION = 140,  
PROP_RECORD_COUNT = 141,  
PROP_START_TIME = 142,  
PROP_STOP_TIME = 143,  
PROP_STOP_WHEN_FULL = 144,  
PROP_TOTAL_RECORD_COUNT = 145,  
PROP_VALID_SAMPLES = 146,  
PROP_WINDOW_INTERVAL = 147,  
PROP_WINDOW_SAMPLES = 148,  
PROP_MAXIMUM_VALUE_TIMESTAMP = 149,  
PROP_MINIMUM_VALUE_TIMESTAMP = 150,  
PROP_VARIANCE_VALUE = 151,  
PROP_ACTIVE_COV_SUBSCRIPTIONS = 152,  
PROP_BACKUP_FAILURE_TIMEOUT = 153,  
PROP_CONFIGURATION_FILES = 154,  
PROP_DATABASE_REVISION = 155,
```

```
PROP_DIRECT_READING = 156,  
PROP_LAST_RESTORE_TIME = 157,  
PROP_MAINTENANCE_REQUIRED = 158,  
PROP_MEMBER_OF = 159,  
PROP_MODE = 160,  
PROP_OPERATION_EXPECTED = 161,  
PROP_SETTING = 162,  
PROP_SILENCED = 163,  
PROP_TRACKING_VALUE = 164,  
PROP_ZONE_MEMBERS = 165,  
PROP_LIFE_SAFETY_ALARM_VALUES = 166,  
PROP_MAX_SEGMENTS_ACCEPTED = 167,  
PROP_PROFILE_NAME = 168,  
PROP_AUTO_SLAVE_DISCOVERY = 169,  
PROP_MANUAL_SLAVE_ADDRESS_BINDING = 170,  
PROP_SLAVE_ADDRESS_BINDING = 171,  
PROP_SLAVE_PROXY_ENABLE = 172,  
PROP_LAST_NOTIFY_TIME = 173,  
PROP_SCHEDULE_DEFAULT = 174,  
PROP_ACCEPTED_MODES = 175,  
PROP_ADJUST_VALUE = 176,  
PROP_COUNT = 177,  
PROP_COUNT_BEFORE_CHANGE = 178,  
PROP_COUNT_CHANGE_TIME = 179,  
PROP_COV_PERIOD = 180,  
PROP_INPUT_REFERENCE = 181,  
PROP_LIMIT_MONITORING_INTERVAL = 182,  
PROP_LOGGING_DEVICE = 183,  
PROP_LOGGING_RECORD = 184,  
PROP_PRESCALE = 185,  
PROP_PULSE_RATE = 186,  
PROP_SCALE = 187,  
PROP_SCALE_FACTOR = 188,  
PROP_UPDATE_TIME = 189,  
PROP_VALUE_BEFORE_CHANGE = 190,  
PROP_VALUE_SET = 191,  
PROP_VALUE_CHANGE_TIME = 192,  
/* enumerations 193-206 are new */  
PROP_ALIGN_INTERVALS = 193,  
PROP_GROUP_MEMBER_NAMES = 194,  
PROP_INTERVAL_OFFSET = 195,  
PROP_LAST_RESTART_REASON = 196,  
PROP_LOGGING_TYPE = 197,  
PROP_MEMBER_STATUS_FLAGS = 198,  
PROP_NOTIFICATION_PERIOD = 199,  
PROP_PREVIOUS_NOTIFY_RECORD = 200,  
PROP_REQUESTED_UPDATE_INTERVAL = 201,  
PROP_RESTART_NOTIFICATION_RECIPIENTS = 202,  
PROP_TIME_OF_DEVICE_RESTART = 203,  
PROP_TIME_SYNCHRONIZATION_INTERVAL = 204,  
PROP_TRIGGER = 205,  
PROP_UTC_TIME_SYNCHRONIZATION_RECIPIENTS = 206,  
/* enumerations 207-211 are used in Addendum d to ANSI/ASHRAE 135-2004 */  
PROP_NODE_SUBTYPE = 207,  
PROP_NODE_TYPE = 208,  
PROP_STRUCTURED_OBJECT_LIST = 209,  
PROP_SUBORDINATE_ANNOTATIONS = 210,  
PROP_SUBORDINATE_LIST = 211,  
/* enumerations 212-225 are used in Addendum e to ANSI/ASHRAE 135-2004 */
```

```
PROP_ACTUAL_SHED_LEVEL = 212,  
PROP_DUTY_WINDOW = 213,  
PROP_EXPECTED_SHED_LEVEL = 214,  
PROP_FULL_DUTY_BASELINE = 215,  
/* enumerations 216-217 are used in Addendum i to ANSI/ASHRAE 135-2004 */  
PROP_BLINK_PRIORITY_THRESHOLD = 216,  
PROP_BLINK_TIME = 217,  
/* enumerations 218-225 are used in Addendum e to ANSI/ASHRAE 135-2004 */  
PROP_REQUESTED_SHED_LEVEL = 218,  
PROP_SHED_DURATION = 219,  
PROP_SHED_LEVEL_DESCRIPTIONS = 220,  
PROP_SHED_LEVELS = 221,  
PROP_STATE_DESCRIPTION = 222,  
/* enumerations 223-225 are used in Addendum i to ANSI/ASHRAE 135-2004 */  
PROP_FADE_TIME = 223,  
PROP_LIGHTING_COMMAND = 224,  
PROP_LIGHTING_COMMAND_PRIORITY = 225,  
/* enumerations 226-235 are used in Addendum f to ANSI/ASHRAE 135-2004 */  
PROP_DOOR_ALARM_STATE = 226,  
PROP_DOOR_EXTENDED_PULSE_TIME = 227,  
PROP_DOOR_MEMBERS = 228,  
PROP_DOOR_OPEN_TOO_LONG_TIME = 229,  
PROP_DOOR_PULSE_TIME = 230,  
PROP_DOOR_STATUS = 231,  
PROP_DOOR_UNLOCK_DELAY_TIME = 232,  
PROP_LOCK_STATUS = 233,  
PROP_MASKED_ALARM_VALUES = 234,  
PROP_SECURED_STATUS = 235,  
/* enumerations 236-243 are used in Addendum i to ANSI/ASHRAE 135-2004 */  
PROP_OFF_DELAY = 236,  
PROP_ON_DELAY = 237,  
PROP_POWER = 238,  
PROP_POWER_ON_VALUE = 239,  
PROP_PROGRESS_VALUE = 240,  
PROP_RAMP_RATE = 241,  
PROP_STEP_INCREMENT = 242,  
PROP_SYSTEM_FAILURE_VALUE = 243,  
/* enumerations 244-311 are used in Addendum j to ANSI/ASHRAE 135-2004 */  
PROP_ABSENTEE_LIMIT = 244,  
PROP_ACCESS_ALARM_EVENTS = 245,  
PROP_ACCESS_DOORS = 246,  
PROP_ACCESS_EVENT = 247,  
PROP_ACCESS_EVENT_AUTHENTICATION_FACTOR = 248,  
PROP_ACCESS_EVENT_CREDENTIAL = 249,  
PROP_ACCESS_EVENT_TIME = 250,  
PROP_ACCESS_RULES = 251,  
PROP_ACCESS_RULES_ENABLE = 252,  
PROP_ACCESS_TRANSACTION_EVENTS = 253,  
PROP_ACCOMPANIED = 254,  
PROP_ACTIVATION_TIME = 255,  
PROP_ACTIVE_AUTHENTICATION_POLICY = 256,  
PROP_ASSIGNED_ACCESS_RIGHTS = 257,  
PROP_AUTHENTICATION_FACTOR_INPUT_LIST = 258,  
PROP_AUTHENTICATION_FACTORS = 259,  
PROP_AUTHENTICATION_POLICY_LIST = 260,  
PROP_AUTHENTICATION_POLICY_NAMES = 261,  
PROP_AUTHORIZATION_MODE = 262,  
PROP_BELONGS_TO = 263,  
PROP_CREDENTIAL_DISABLE = 264,
```

```
PROP_CREDENTIAL_STATUS = 265,  
PROP_CREDENTIALS = 266,  
PROP_CREDENTIALS_IN_ZONE = 267,  
PROP_DAYS_REMAINING = 268,  
PROP_ENTRY_POINTS = 269,  
PROP_EXIT_POINTS = 270,  
PROP_EXPIRY_TIME = 271,  
PROP_EXTENDED_TIME_ENABLE = 272,  
PROP_FAILED_ATTEMPT_EVENTS = 273,  
PROP_FAILED_ATTEMPTS = 274,  
PROP_FAILED_ATTEMPTS_TIME = 275,  
PROP_FORMAT_CLASS_SUPPORTED = 276,  
PROP_FORMAT_TYPE = 277,  
PROP_LAST_ACCESS_EVENT = 278,  
PROP_LAST_ACCESS_POINT = 279,  
PROP_LAST_CREDENTIAL_ADDED = 280,  
PROP_LAST_CREDENTIAL_ADDED_TIME = 281,  
PROP_LAST_CREDENTIAL_REMOVED = 282,  
PROP_LAST_CREDENTIAL_REMOVED_TIME = 283,  
PROP_LAST_USE_TIME = 284,  
PROP_LOCKDOWN = 285,  
PROP_LOCKDOWN_RELINQUISH_TIME = 286,  
PROP_MASTER_EXEMPTION = 287,  
PROP_MAX_FAILED_ATTEMPTS = 288,  
PROP_MEMBERS = 289,  
PROP_MASTER_POINT = 290,  
PROP_NUMBER_OF_AUTHENTICATION_POLICIES = 291,  
PROP_OCCUPANCY_COUNT = 293,  
PROP_OCCUPANCY_COUNT_ENABLE = 294,  
PROP_OCCUPANCY_COUNT_EXEMPTION = 295,  
PROP_OCCUPANCY_LOWER_THRESHOLD = 296,  
PROP_OCCUPANCY_LOWER_THRESHOLD_ENFORCED = 297,  
PROP_OCCUPANCY_STATE = 298,  
PROP_OCCUPANCY_UPPER_LIMIT = 299,  
PROP_OCCUPANCY_UPPER_LIMIT_ENFORCED = 300,  
PROP_PASSBACK_EXEMPTION = 301,  
PROP_PASSBACK_MODE = 302,  
PROP_PASSBACK_TIMEOUT = 303,  
PROP_POSITIVE_ACCESS_RULES = 304,  
PROP_READ_STATUS = 305,  
PROP_REASON_FOR_DISABLE = 306,  
PROP_THREAT_AUTHORITY = 307,  
PROP_THREAT_LEVEL = 308,  
PROP_TRACE_FLAG = 309,  
PROP_TRANSACTION_NOTIFICATION_CLASS = 310,  
PROP_USER_EXTERNAL_IDENTIFIER = 311,  
/* enumerations 312-313 are used in Addendum k to ANSI/ASHRAE 135-2004 */  
PROP_CHARACTER_SET = 312,  
PROP_STRICT_CHARACTER_MODE = 313,  
/* enumerations 312-313 are used in Addendum k to ANSI/ASHRAE 135-2004 */  
PROP_BACKUP_AND_RESTORE_STATE = 314,  
PROP_BACKUP_PREPARATION_TIME = 315,  
PROP_RESTORE_PREPARATION_TIME = 316,  
/* enumerations 317-323 are used in Addendum j to ANSI/ASHRAE 135-2004 */  
PROP_USER_INFORMATION_REFERENCE = 317,  
PROP_USER_NAME = 318,  
PROP_USER_TYPE = 319,  
PROP_USES_REMAINING = 320,  
PROP_VENDOR_FORMAT_IDENTIFIER = 321,
```

```

PROP_ZONE_FROM = 322,
PROP_ZONE_TO = 323,
/* enumerations 324-325 are used in Addendum i to ANSI/ASHRAE 135-2004 */
PROP_BINARY_ACTIVE_VALUE = 324,
PROP_BINARY_INACTIVE_VALUE = 325
    /* The special property identifiers all, optional, and required */
    /* are reserved for use in the ReadPropertyConditional and */
    /* ReadPropertyMultiple services or services not defined in this standard. */
    /* Enumerated values 0-511 are reserved for definition by ASHRAE. */
    /* Enumerated values 512-4194303 may be used by others subject to the */
    /* procedures and constraints described in Clause 23. */
} BACNET_PROPERTY_ID;

```

## E.2 BACnet Engineering Units Codes

BACnet engineering units codes may be found in your copy of the BACnet protocol specification, ANSI/ASHRAE Standard 135. That document is copyrighted, but the C enumeration shown below for reference is taken from open source code available under GPL at <http://sourceforge.net>, and provides essentially the same information (copyrighted by Steve Karg, licensed under GPL as noted at <http://sourceforge.net>).

```

typedef enum {
    /* Acceleration */
    UNITS_METERS_PER_SECOND_PER_SECOND = 166,
    /* Area */
    UNITS_SQUARE_METERS = 0,
    UNITS_SQUARE_CENTIMETERS = 116,
    UNITS_SQUARE_FEET = 1,
    UNITS_SQUARE_INCHES = 115,
    /* Currency */
    UNITS_CURRENCY1 = 105,
    UNITS_CURRENCY2 = 106,
    UNITS_CURRENCY3 = 107,
    UNITS_CURRENCY4 = 108,
    UNITS_CURRENCY5 = 109,
    UNITS_CURRENCY6 = 110,
    UNITS_CURRENCY7 = 111,
    UNITS_CURRENCY8 = 112,
    UNITS_CURRENCY9 = 113,
    UNITS_CURRENCY10 = 114,
    /* Electrical */
    UNITS_MILLIAMPERES = 2,
    UNITS_AMPERES = 3,
    UNITS_AMPERES_PER_METER = 167,
    UNITS_AMPERES_PER_SQUARE_METER = 168,
    UNITS_AMPERE_SQUARE_METERS = 169,
    UNITS_FARADS = 170,
    UNITS_HENRYS = 171,
    UNITS_OHMS = 4,
    UNITS_OHM_METERS = 172,
    UNITS_MILLIOHMS = 145,
    UNITS_KILOHMS = 122,
    UNITS_MEGOHMS = 123,
    UNITS_SIEMENS = 173, /* 1 mho equals 1 siemens */
    UNITS_SIEMENS_PER_METER = 174,
    UNITS_TESLAS = 175,
    UNITS_VOLTS = 5,
    UNITS_MILLIVOLTS = 124,
    UNITS_KILOVOLTS = 6,

```

```
UNITS_MEGAVOLTS = 7,  
UNITS_VOLT_AMPERES = 8,  
UNITS_KILOVOLT_AMPERES = 9,  
UNITS_MEGAVOLT_AMPERES = 10,  
UNITS_VOLT_AMPERES_REACTIVE = 11,  
UNITS_KILOVOLT_AMPERES_REACTIVE = 12,  
UNITS_MEGAVOLT_AMPERES_REACTIVE = 13,  
UNITS_VOLTS_PER_DEGREE_KELVIN = 176,  
UNITS_VOLTS_PER_METER = 177,  
UNITS_DEGREES_PHASE = 14,  
UNITS_POWER_FACTOR = 15,  
UNITS_WEBERS = 178,  
/* Energy */  
UNITS_JOULES = 16,  
UNITS_KILOJOULES = 17,  
UNITS_KILOJOULES_PER_KILOGRAM = 125,  
UNITS_MEGAJOULES = 126,  
UNITS_WATT_HOURS = 18,  
UNITS_KILOWATT_HOURS = 19,  
UNITS_MEGAWATT_HOURS = 146,  
UNITS_BTUS = 20,  
UNITS_KILO_BTUS = 147,  
UNITS_MEGA_BTUS = 148,  
UNITS_THERMS = 21,  
UNITS_TON_HOURS = 22,  
/* Enthalpy */  
UNITS_JOULES_PER_KILOGRAM_DRY_AIR = 23,  
UNITS_KILOJOULES_PER_KILOGRAM_DRY_AIR = 149,  
UNITS_MEGAJOULES_PER_KILOGRAM_DRY_AIR = 150,  
UNITS_BTUS_PER_POUND_DRY_AIR = 24,  
UNITS_BTUS_PER_POUND = 117,  
/* Entropy */  
UNITS_JOULES_PER_DEGREE_KELVIN = 127,  
UNITS_KILOJOULES_PER_DEGREE_KELVIN = 151,  
UNITS_MEGAJOULES_PER_DEGREE_KELVIN = 152,  
UNITS_JOULES_PER_KILOGRAM_DEGREE_KELVIN = 128,  
/* Force */  
UNITS_NEWTON = 153,  
/* Frequency */  
UNITS_CYCLES_PER_HOUR = 25,  
UNITS_CYCLES_PER_MINUTE = 26,  
UNITS_HERTZ = 27,  
UNITS_KILOHERTZ = 129,  
UNITS_MEGAHERTZ = 130,  
UNITS_PER_HOUR = 131,  
/* Humidity */  
UNITS_GRAMS_OF_WATER_PER_KILOGRAM_DRY_AIR = 28,  
UNITS_PERCENT_RELATIVE_HUMIDITY = 29,  
/* Length */  
UNITS_MILLIMETERS = 30,  
UNITS_CENTIMETERS = 118,  
UNITS_METERS = 31,  
UNITS_INCHES = 32,  
UNITS_FEET = 33,  
/* Light */  
UNITS_CANDELAS = 179,  
UNITS_CANDELAS_PER_SQUARE_METER = 180,  
UNITS_WATTS_PER_SQUARE_FOOT = 34,  
UNITS_WATTS_PER_SQUARE_METER = 35,
```

```
UNITS_LUMENS = 36,  
UNITS_LUXES = 37,  
UNITS_FOOT_CANDLES = 38,  
/* Mass */  
UNITS_KILOGRAMS = 39,  
UNITS_POUNDS_MASS = 40,  
UNITS_TONS = 41,  
/* Mass Flow */  
UNITS_GRAMS_PER_SECOND = 154,  
UNITS_GRAMS_PER_MINUTE = 155,  
UNITS_KILOGRAMS_PER_SECOND = 42,  
UNITS_KILOGRAMS_PER_MINUTE = 43,  
UNITS_KILOGRAMS_PER_HOUR = 44,  
UNITS_POUNDS_MASS_PER_SECOND = 119,  
UNITS_POUNDS_MASS_PER_MINUTE = 45,  
UNITS_POUNDS_MASS_PER_HOUR = 46,  
UNITS_TONS_PER_HOUR = 156,  
/* Power */  
UNITS_MILLIWATTS = 132,  
UNITS_WATTS = 47,  
UNITS_KILOWATTS = 48,  
UNITS_MEGAWATTS = 49,  
UNITS_BTUS_PER_HOUR = 50,  
UNITS_KILO_BTUS_PER_HOUR = 157,  
UNITS_HORSEPOWER = 51,  
UNITS_TONS_REFRIGERATION = 52,  
/* Pressure */  
UNITS_PASCALS = 53,  
UNITS_HECTOPASCALS = 133,  
UNITS_KILOPASCALS = 54,  
UNITS_MILLIBARS = 134,  
UNITS_BARS = 55,  
UNITS_POUNDS_FORCE_PER_SQUARE_INCH = 56,  
UNITS_CENTIMETERS_OF_WATER = 57,  
UNITS_INCHES_OF_WATER = 58,  
UNITS_MILLIMETERS_OF_MERCURY = 59,  
UNITS_CENTIMETERS_OF_MERCURY = 60,  
UNITS_INCHES_OF_MERCURY = 61,  
/* Temperature */  
UNITS_DEGREES_CELSIUS = 62,  
UNITS_DEGREES_KELVIN = 63,  
UNITS_DEGREES_KELVIN_PER_HOUR = 181,  
UNITS_DEGREES_KELVIN_PER_MINUTE = 182,  
UNITS_DEGREES_FAHRENHEIT = 64,  
UNITS_DEGREE_DAYS_CELSIUS = 65,  
UNITS_DEGREE_DAYS_FAHRENHEIT = 66,  
UNITS_DELTA_DEGREES_FAHRENHEIT = 120,  
UNITS_DELTA_DEGREES_KELVIN = 121,  
/* Time */  
UNITS_YEARS = 67,  
UNITS_MONTHS = 68,  
UNITS_WEEKS = 69,  
UNITS_DAYS = 70,  
UNITS_HOURS = 71,  
UNITS_MINUTES = 72,  
UNITS_SECONDS = 73,  
UNITS_HUNDREDTHS_SECONDS = 158,  
UNITS_MILLISECONDS = 159,  
/* Torque */
```



```
UNITS_NEWTON_METERS = 160,  
/* Velocity */  
UNITS_MILLIMETERS_PER_SECOND = 161,  
UNITS_MILLIMETERS_PER_MINUTE = 162,  
UNITS_METERS_PER_SECOND = 74,  
UNITS_METERS_PER_MINUTE = 163,  
UNITS_METERS_PER_HOUR = 164,  
UNITS_KILOMETERS_PER_HOUR = 75,  
UNITS_FEET_PER_SECOND = 76,  
UNITS_FEET_PER_MINUTE = 77,  
UNITS_MILES_PER_HOUR = 78,  
/* Volume */  
UNITS_CUBIC_FEET = 79,  
UNITS_CUBIC_METERS = 80,  
UNITS_IMPERIAL_GALLONS = 81,  
UNITS_LITERS = 82,  
UNITS_US_GALLONS = 83,  
/* Volumetric Flow */  
UNITS_CUBIC_FEET_PER_SECOND = 142,  
UNITS_CUBIC_FEET_PER_MINUTE = 84,  
UNITS_CUBIC_METERS_PER_SECOND = 85,  
UNITS_CUBIC_METERS_PER_MINUTE = 165,  
UNITS_CUBIC_METERS_PER_HOUR = 135,  
UNITS_IMPERIAL_GALLONS_PER_MINUTE = 86,  
UNITS_LITERS_PER_SECOND = 87,  
UNITS_LITERS_PER_MINUTE = 88,  
UNITS_LITERS_PER_HOUR = 136,  
UNITS_US_GALLONS_PER_MINUTE = 89,  
/* Other */  
UNITS_DEGREES_ANGULAR = 90,  
UNITS_DEGREES_CELSIUS_PER_HOUR = 91,  
UNITS_DEGREES_CELSIUS_PER_MINUTE = 92,  
UNITS_DEGREES_FAHRENHEIT_PER_HOUR = 93,  
UNITS_DEGREES_FAHRENHEIT_PER_MINUTE = 94,  
UNITS_JOULE_SECONDS = 183,  
UNITS_KILOGRAMS_PER_CUBIC_METER = 186,  
UNITS_KW_HOURS_PER_SQUARE_METER = 137,  
UNITS_KW_HOURS_PER_SQUARE_FOOT = 138,  
UNITS_MEGAJOULES_PER_SQUARE_METER = 139,  
UNITS_MEGAJOULES_PER_SQUARE_FOOT = 140,  
UNITS_NO_UNITS = 95,  
UNITS_NEWTON_SECONDS = 187,  
UNITS_NEWTONS_PER_METER = 188,  
UNITS_PARTS_PER_MILLION = 96,  
UNITS_PARTS_PER_BILLION = 97,  
UNITS_PERCENT = 98,  
UNITS_PERCENT_OBSCURATION_PER_FOOT = 143,  
UNITS_PERCENT_OBSCURATION_PER_METER = 144,  
UNITS_PERCENT_PER_SECOND = 99,  
UNITS_PER_MINUTE = 100,  
UNITS_PER_SECOND = 101,  
UNITS_PSI_PER_DEGREE_FAHRENHEIT = 102,  
UNITS_RADIANS = 103,  
UNITS_RADIANS_PER_SECOND = 184,  
UNITS_REVOLUTIONS_PER_MINUTE = 104,  
UNITS_SQUARE_METERS_PER_NEWTON = 185,  
UNITS_WATTS_PER_METER_PER_DEGREE_KELVIN = 189,  
UNITS_WATTS_PER_SQUARE_METER_DEGREE_KELVIN = 141,  
; /* Enumerated values 0-255 are reserved for definition by ASHRAE. */
```

```

    /* Enumerated values 256-65535 may be used by others subject to */
    /* the procedures and constraints described in Clause 23. */
    /* The last enumeration used in this version is 189. */
    MAX_UNITS = 190
} BACNET_ENGINEERING_UNITS;

```

### E.3 BACnet Error Codes

BACnet error codes may be found in your copy of the BACnet protocol specification, ANSI/ASHRAE Standard 135. That document is copyrighted, but the C enumeration shown below for reference is taken from open source code available under GPL at <http://sourceforge.net>, and provides essentially the same information (copyrighted by Steve Karg, licensed under GPL as noted at <http://sourceforge.net>).

BACnet error codes actually consist of two codes: The error Class and the error Code. The classes are the shorter list shown first below, and the codes that may occur for each class follow.

```

typedef enum {
    ERROR_CLASS_DEVICE = 0,
    ERROR_CLASS_OBJECT = 1,
    ERROR_CLASS_PROPERTY = 2,
    ERROR_CLASS_RESOURCES = 3,
    ERROR_CLASS_SECURITY = 4,
    ERROR_CLASS_SERVICES = 5,
    ERROR_CLASS_VT = 6,
    ERROR_CLASS_COMMUNICATION = 7,
    /* Enumerated values 0-63 are reserved for definition by ASHRAE. */
    /* Enumerated values 64-65535 may be used by others subject to */
    /* the procedures and constraints described in Clause 23. */
    MAX_BACNET_ERROR_CLASS = 8
} BACNET_ERROR_CLASS;

```

```

/* These are sorted in the order given in
Clause 18. ERROR, REJECT AND ABORT CODES
The Class and Code pairings are required
to be used in accordance with Clause 18. */

```

```

typedef enum {
/* valid for all classes */
    ERROR_CODE_OTHER = 0,

```

```

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

```

```

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

```

```

/* Error Class - Property */
    ERROR_CODE_CHARACTER_SET_NOT_SUPPORTED = 41,
    ERROR_CODE_DATATYPE_NOT_SUPPORTED = 47,
    ERROR_CODE_INCONSISTENT_SELECTION_CRITERION = 8,
    ERROR_CODE_INVALID_ARRAY_INDEX = 42,

```

```
ERROR_CODE_INVALID_DATA_TYPE = 9,  
ERROR_CODE_NOT_COV_PROPERTY = 44,  
ERROR_CODE_OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED = 45,  
ERROR_CODE_PROPERTY_IS_NOT_AN_ARRAY = 50,  
/* ERROR_CODE_READ_ACCESS_DENIED = 27, */  
ERROR_CODE_UNKNOWN_PROPERTY = 32,  
ERROR_CODE_VALUE_OUT_OF_RANGE = 37,  
ERROR_CODE_WRITE_ACCESS_DENIED = 40,  
  
/* Error Class - Resources */  
ERROR_CODE_NO_SPACE_FOR_OBJECT = 18,  
ERROR_CODE_NO_SPACE_TO_ADD_LIST_ELEMENT = 19,  
ERROR_CODE_NO_SPACE_TO_WRITE_PROPERTY = 20,  
  
/* Error Class - Security */  
ERROR_CODE_AUTHENTICATION_FAILED = 1,  
/* ERROR_CODE_CHARACTER_SET_NOT_SUPPORTED = 41, */  
ERROR_CODE_INCOMPATIBLE_SECURITY_LEVELS = 6,  
ERROR_CODE_INVALID_OPERATOR_NAME = 12,  
ERROR_CODE_KEY_GENERATION_ERROR = 15,  
ERROR_CODE_PASSWORD_FAILURE = 26,  
ERROR_CODE_SECURITY_NOT_SUPPORTED = 28,  
ERROR_CODE_TIMEOUT = 30,  
  
/* Error Class - Services */  
/* ERROR_CODE_CHARACTER_SET_NOT_SUPPORTED = 41, */  
ERROR_CODE_COV_SUBSCRIPTION_FAILED = 43,  
ERROR_CODE_DUPLICATE_NAME = 48,  
ERROR_CODE_DUPLICATE_OBJECT_ID = 49,  
ERROR_CODE_FILE_ACCESS_DENIED = 5,  
ERROR_CODE_INCONSISTENT_PARAMETERS = 7,  
ERROR_CODE_INVALID_CONFIGURATION_DATA = 46,  
ERROR_CODE_INVALID_FILE_ACCESS_METHOD = 10,  
ERROR_CODE_ERROR_CODE_INVALID_FILE_START_POSITION = 11,  
ERROR_CODE_INVALID_PARAMETER_DATA_TYPE = 13,  
ERROR_CODE_INVALID_TIME_STAMP = 14,  
ERROR_CODE_MISSING_REQUIRED_PARAMETER = 16,  
/* ERROR_CODE_OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED = 45, */  
ERROR_CODE_PROPERTY_IS_NOT_A_LIST = 22,  
ERROR_CODE_SERVICE_REQUEST_DENIED = 29,  
  
/* Error Class - VT */  
ERROR_CODE_UNKNOWN_VT_CLASS = 34,  
ERROR_CODE_UNKNOWN_VT_SESSION = 35,  
ERROR_CODE_NO_VT_SESSIONS_AVAILABLE = 21,  
ERROR_CODE_VT_SESSION_ALREADY_CLOSED = 38,  
ERROR_CODE_VT_SESSION_TERMINATION_FAILURE = 39,  
  
/* unused */  
ERROR_CODE_RESERVED1 = 33,  
/* new error codes from new addenda */  
ERROR_CODE_ABORT_BUFFER_OVERFLOW = 51,  
ERROR_CODE_ABORT_INVALID_APDU_IN_THIS_STATE = 52,  
ERROR_CODE_ABORT_PREEMPTED_BY_HIGHER_PRIORITY_TASK = 53,  
ERROR_CODE_ABORT_SEGMENTATION_NOT_SUPPORTED = 54,  
ERROR_CODE_ABORT_PROPRIETARY = 55,  
ERROR_CODE_ABORT_OTHER = 56,  
ERROR_CODE_INVALID_TAG = 57,  
ERROR_CODE_NETWORK_DOWN = 58,
```

```
ERROR_CODE_REJECT_BUFFER_OVERFLOW = 59,  
ERROR_CODE_REJECT_INCONSISTENT_PARAMETERS = 60,  
ERROR_CODE_REJECT_INVALID_PARAMETER_DATA_TYPE = 61,  
ERROR_CODE_REJECT_INVALID_TAG = 62,  
ERROR_CODE_REJECT_MISSING_REQUIRED_PARAMETER = 63,  
ERROR_CODE_REJECT_PARAMETER_OUT_OF_RANGE = 64,  
ERROR_CODE_REJECT_TOO_MANY_ARGUMENTS = 65,  
ERROR_CODE_REJECT_UNDEFINED_ENUMERATION = 66,  
ERROR_CODE_REJECT_UNRECOGNIZED_SERVICE = 67,  
ERROR_CODE_REJECT_PROPRIETARY = 68,  
ERROR_CODE_REJECT_OTHER = 69,  
ERROR_CODE_UNKNOWN_DEVICE = 70,  
ERROR_CODE_UNKNOWN_ROUTE = 71,  
ERROR_CODE_VALUE_NOT_INITIALIZED = 72,  
ERROR_CODE_INVALID_EVENT_STATE = 73,  
ERROR_CODE_NO_ALARM_CONFIGURED = 74,  
ERROR_CODE_LOG_BUFFER_FULL = 75,  
ERROR_CODE_LOGGED_VALUE_PURGED = 76,  
ERROR_CODE_NO_PROPERTY_SPECIFIED = 77,  
ERROR_CODE_NOT_CONFIGURED_FOR_TRIGGERED_LOGGING = 78,  
ERROR_CODE_UNKNOWN_SUBSCRIPTION = 79,  
ERROR_CODE_PARAMETER_OUT_OF_RANGE = 80,  
ERROR_CODE_UNKNOWN_FILE_SIZE = 81,  
ERROR_CODE_BUSY = 82,  
ERROR_CODE_COMMUNICATION_DISABLED = 83,  
MAX_BACNET_ERROR_CODE = 84  
/* Enumerated values 0-255 are reserved for definition by ASHRAE. */  
/* Enumerated values 256-65535 may be used by others subject to */  
/* the procedures and constraints described in Clause 23. */  
/* The last enumeration used in this version is 50. */  
} BACNET_ERROR_CODE;
```



## Appendix F USB Driver Installation

### F.1 Driver Installation

DO NOT connect the MTX002 or other Control Solutions USB device prior to installing the drivers. Install drivers first, then plug the device in.

Control Solutions has licensed a USB driver and installer from Thesycon in Germany. 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 everything from Windows XP (32-bit and 64-bit) to Windows 11 and counting. It includes the necessary driver signing verified through Verisign (which became Symantec, then DigiCert).

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:

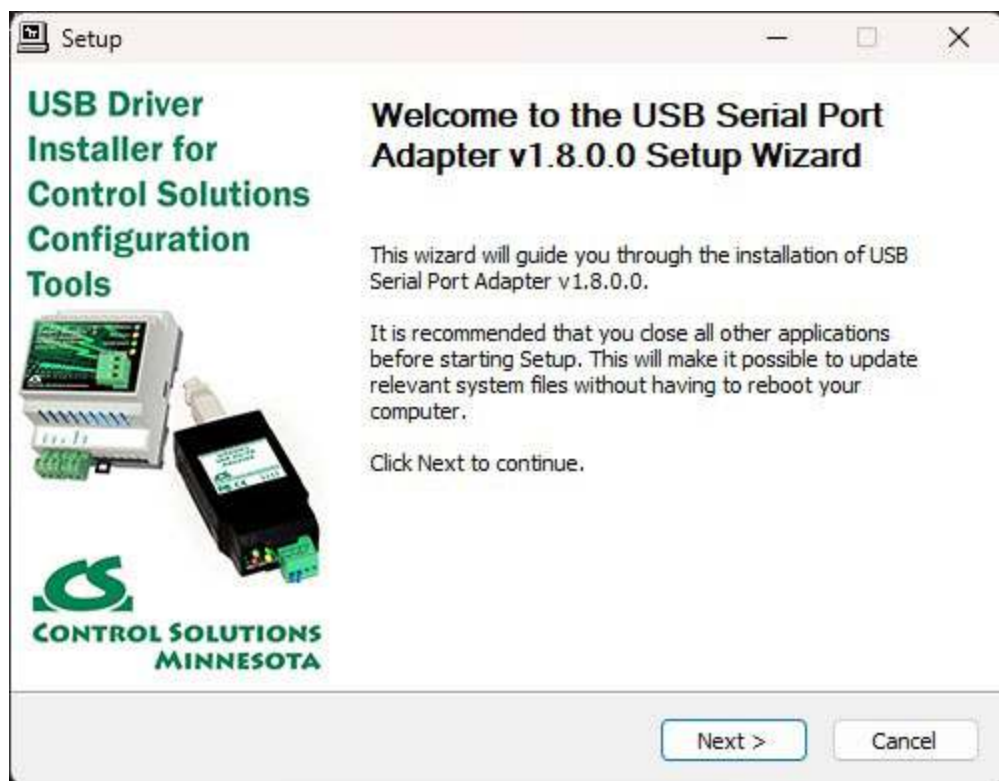


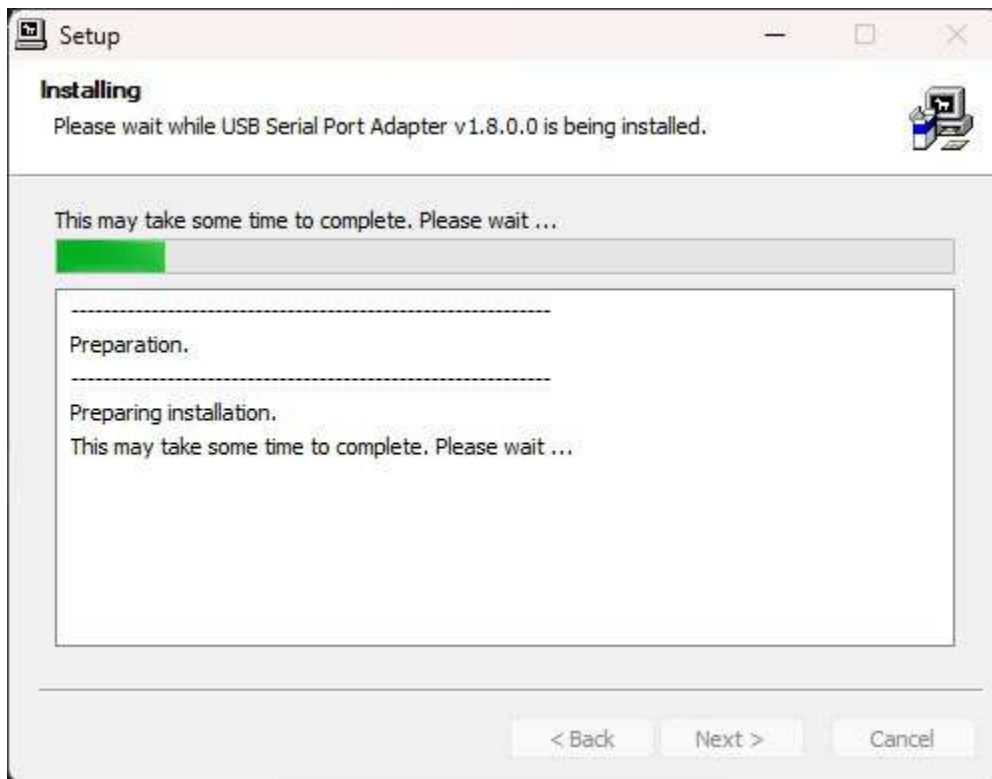
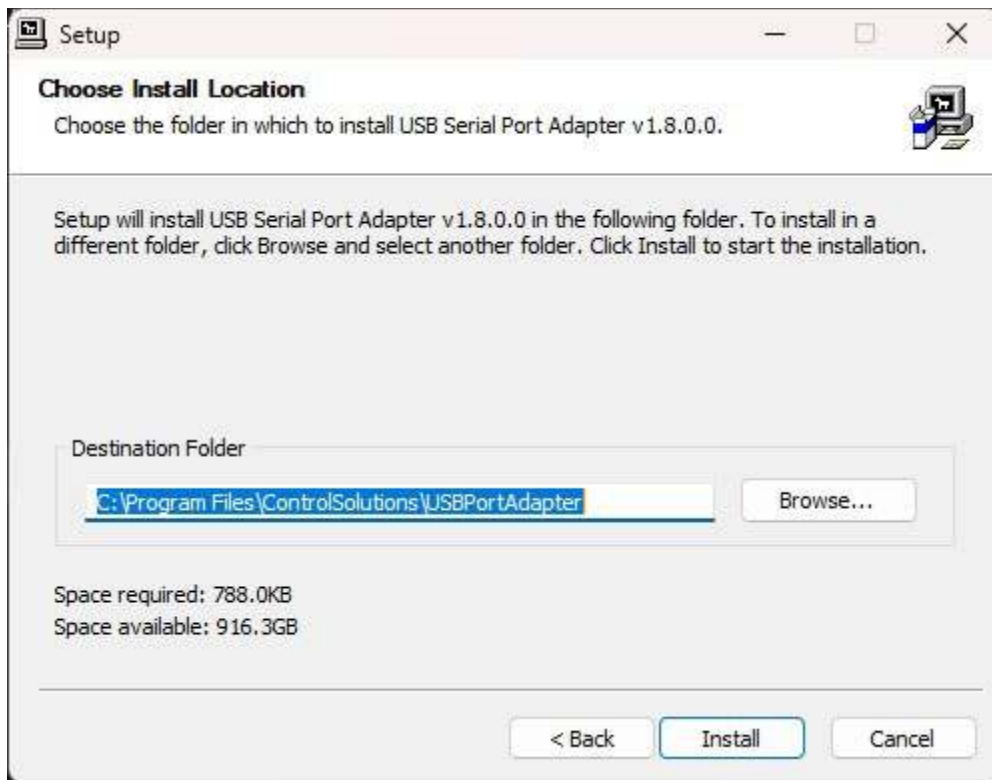
Double click "setup.exe". Say "yes" to any questions about whether to trust this software.

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.

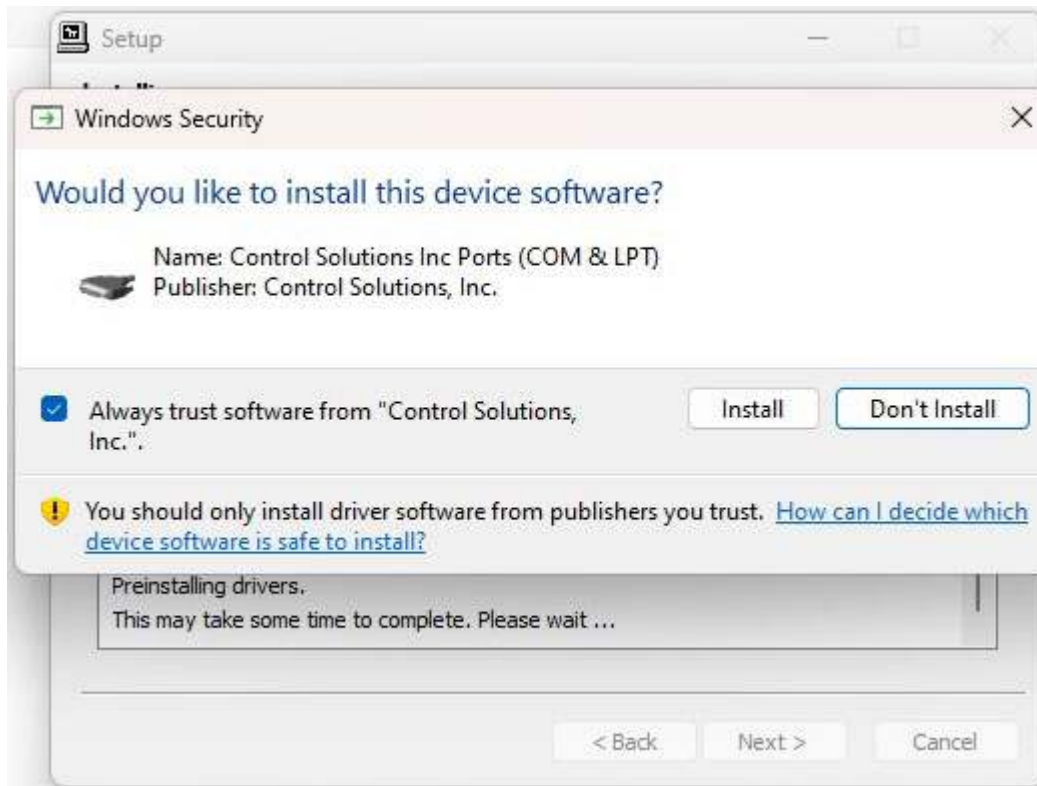
Technically, what you are doing in the process illustrated in the screen shots below is "driver pre-installation". After initial installation of this package, the device will automatically find the right driver when you plug it in, and driver installation will be finalized.

You will most likely get some variation of the "allow this app" warning. Click Yes to proceed.

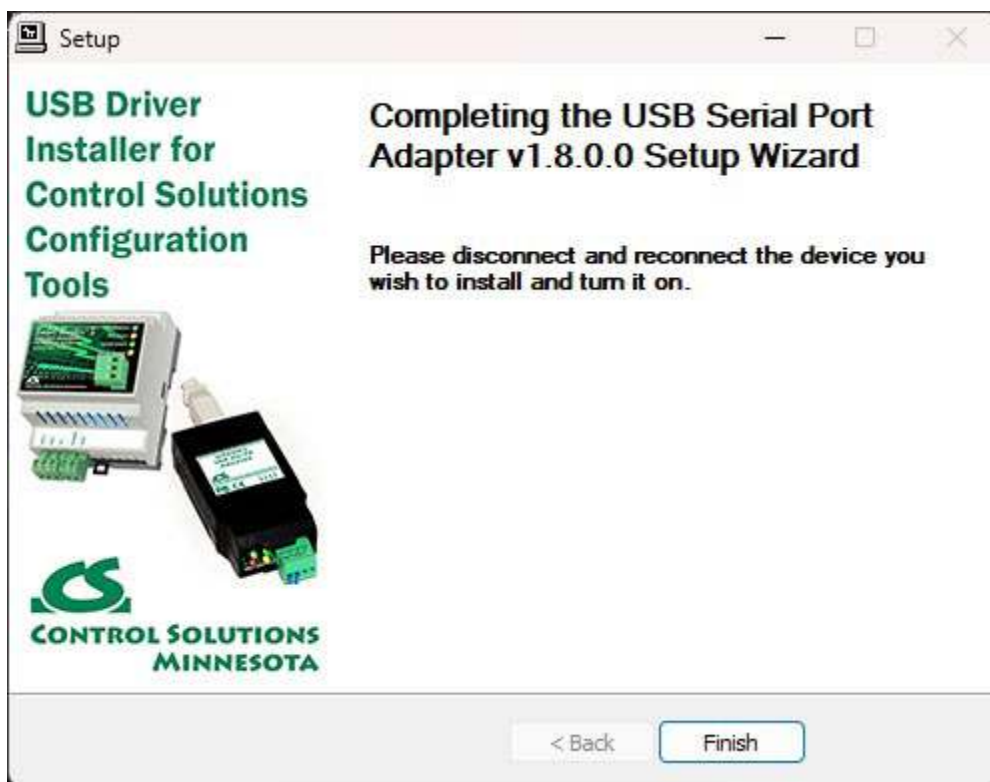




Click on "Install" when you get to this window:



When you get to this screen, you're done. Now plug in your USB device (MTX002), 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.



Ideally you had not already connected the device. This prompt says you may now connect the device and the driver installation will be finalized.



