

Managing a BACnet Setpoint via a Gateway

Configuring a gateway to simply read a setpoint will let you look at the setpoint, but not change it. Configuring a gateway to simply write a setpoint periodically will let you control the setpoint, but not allow anybody or anything else to also be able to change it. In most practical applications, you want to read the setpoint first, apply some decision making about whether it needs to be changed, and write to the setpoint object only if a change is needed. You also want the ability to let other things (e.g. a local operator panel) to make setpoint changes that you observe without attempting to override.

Allowing the proper treatment of a setpoint object was taken into account in the design of the Babel Buster gateway. As defined by BACnet protocol, Input objects are for reading information from some device, and Output objects are for writing information to some device. But the Value object can go in either direction, and the gateway takes advantage of this fact to allow proper handling of setpoints via the gateway.

Within the Babel Buster gateway, Input objects can only be configured to read information from some other device. Output objects can only be configured to write information to some other device. Both the Input and Output objects will have only one remote data mapping, or read/write rule, associated with them. The Value object can both read and write information in a remote device, and is allowed to have two remote data mappings, one to read and another to write the remote device.

The configuration allowing an Analog Output in a remote device, in this case a ValuPoint VP4-2330, to be treated as a setpoint is illustrated here using the BB2-7010 gateway. We will use AV 3 in the gateway to interact with AO 1 in the ValuPoint controller.

Babel Buster 2
BACNET-MODBUS
NETWORK GATEWAY
MODEL BB2-7010-01

CONTROL SOLUTIONS, INC.
MINNESOTA

Data Objects | Modbus | BACnet | System Setup

Analog | Binary | Multi-State

Input Objects | Output Objects | Value Objects

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

Analog Value # 3 Update < Prev Next >

Reliability: 0 Status: 0,0,0,0 Device Link: [BIP R4](#) [BIP W2](#) Out of Service: Deconfigure:

Object name: AO on VP4-2330 Force Present Value: 5.00000

Description: Analog Output managed as setpoint

COV increment: 0.00000 Units: no_units

The Read Map is pretty straight forward. You simply configure the gateway to periodically read AO 1 from the VP4-2330 and save its Present Value in the gateway's own AV 3.

Babel Buster 2
BACNET-MODBUS
NETWORK GATEWAY
MODEL BB2-7010-01

CONTROL SOLUTIONS, INC.
MINNESOTA

Data Objects | Modbus | BACnet | System Setup

BACnet IP Client | Diagnostics

Devices | Client Read Map | Client Write Map

This page creates a map entry that reads data from a remote BACnet IP server for processing here.

Map # 4 Update < Prev Next >

Read property: 85 instance #: 1 of object type: Analog Output

Read from device: VP4-2330 using index: 0

Then apply scale: 0.00000 and offset: 0.00000

Save in local object: AV 3 named: AO on VP4-2330 Repeat this process every: 2.0 seconds.

Apply this default value: 0.00000 after 0 read failure(s).

Client Read Maps Enabled: 5 Insert Delete

The Write Map in the BB2-7010 is configured to take whatever value is in the gateway's own AV 3 and send that to the VP4-2330 controller's AO 1. But there are some things you need to pay attention to here. The two most important ones are marked in red on the screen shot. These two selections will cause AV 3 to be written to the remote AO 1 only when AV 3 is updated.

The "changed by" value can be any value you want to set the threshold at, but zero will mean any update of AV 3 even if there is no actual change in value. If there is no change in value, the AV object must have been explicitly written to using the same value. Simply finding the same value will not result in sending the value when the check box marked in red is selected.

The second item marked in red is the "repeat" option. If "at least" is selected, then it becomes a periodic update, and you don't want that. Select "no more than", and in this case the time value can be zero or any other number you wish to limit update rate to.

The other item to pay attention to if the object you are writing to in the remote device is an Output object, which is required to be commandable by BACnet protocol, is the priority level. In our example, we are using priority 4. If something else has written to the same Output object with a higher (lower numbered) priority, our desired value will be retained by the Output object, but not applied until the higher priority has relinquished the object. If your attempt to write to the Output object appears unsuccessful at first, check to see if anything at a higher priority has written to the same object.

Babel Buster 2
BACNET-MODBUS
NETWORK GATEWAY
MODEL BB2-7010-01

CONTROL SOLUTIONS, INC.
MINNESOTA

Data Objects | Modbus | **BACnet** | System Setup

BACnet IP Client | Diagnostics

Devices | Client Read Map | Client Write Map

This page creates a map entry that writes data to one or more remote BACnet IP servers from data contained here.

Map # 2 Update < Prev Next >

Read local object AV 3 named AO on VP4-2330

Apply default value of 0.00000 at power-up and/or when 0.0 seconds have elapsed with no host update.

Write remote register any time local object has changed by 0.00000 or when 0.0 seconds have elapsed with no change.

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

Apply scale: 0.00000 and offset: 0.00000 Then, using index 0 and priority 4 proceed to

Write property 85 encoded as data type Real

Write to instance # 1 of object type Analog Output at device VP4-2330

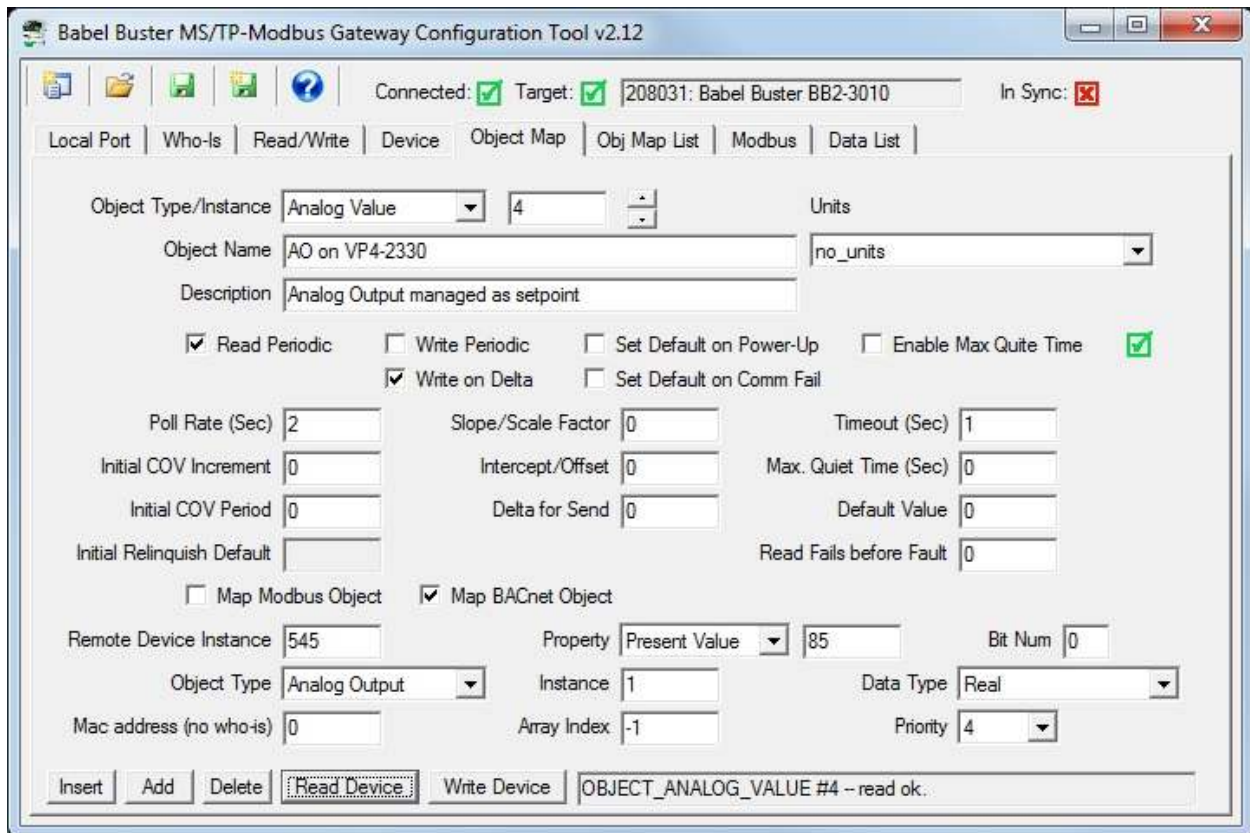
Repeat this process at least no more than every 0.0 seconds.

Client Write Maps Enabled: 3 Insert Delete

The BB2-7010 was used in the above example, but any Babel Buster with a BACnet IP Client will work the same as this example. The same effect can be accomplished in the MS/TP gateway, the Babel Buster BB2-3010 or BB2-3060. Our MS/TP example uses AV 4 for the same purpose as AV 3 in the IP example. The AV number is of no consequence here - the other AV's were already in use in the gateways used to create these examples.

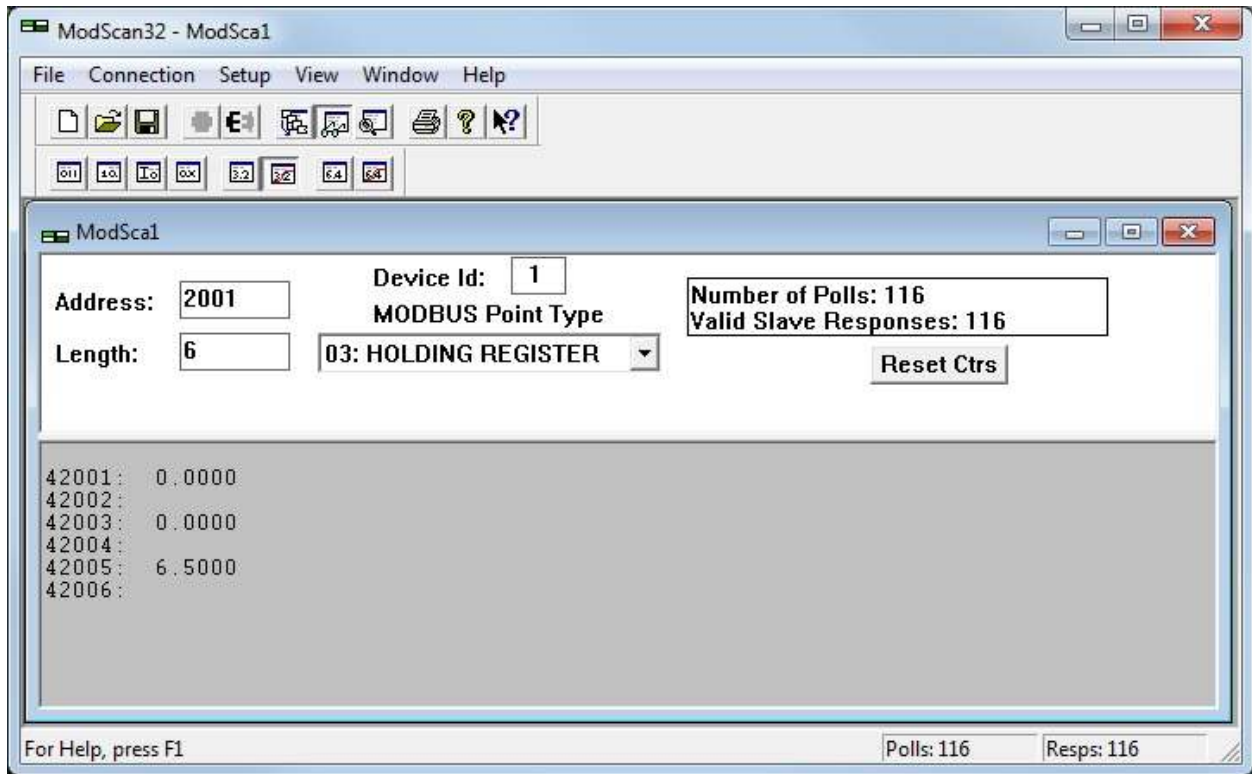
Mapping in the MS/TP gateways is handled a little differently than in the IP gateways. To configure the BB2-3010 to function in the manner we have discussed, select both Read Periodic and Write on Delta for the same AV. Only the AV objects allow you to select both read and write simultaneously in the BB2-3010.

As with the example above, pay attention to priority level when writing to a commandable Output object.

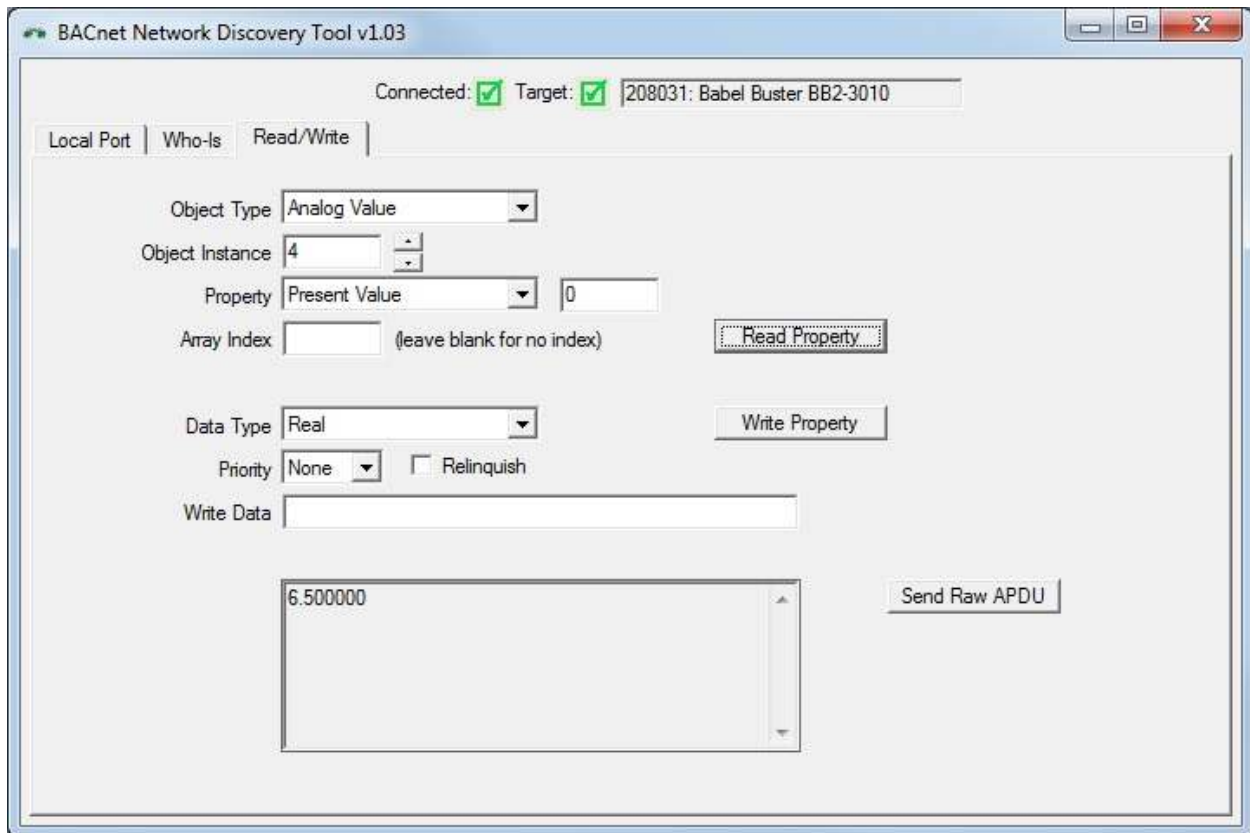


Of course, the whole idea of using one of these gateways is that you want to control a BACnet device from a non-BACnet device, such as a Modbus PLC. To test our example setup, we are using ModScan to emulate the PLC. ModScan is simply polling the holding registers that equate to the AV objects in the BB2-7010 above. Any time we double click on Modbus register 42005, ModScan lets us enter a new value as if the PLC was about to write to that register. Register 42005 correlates with AV 3 in the BB2-7010 above. Any time we change register 42005, the new value shows up in both gateways. This is because the BB2-7010 sees an update to its AV 3, writes the new value to the VP4-2330 controller's AO 1, and then the BB2-3010 sees the updated value in AO 1 and sets its AV 4 accordingly.

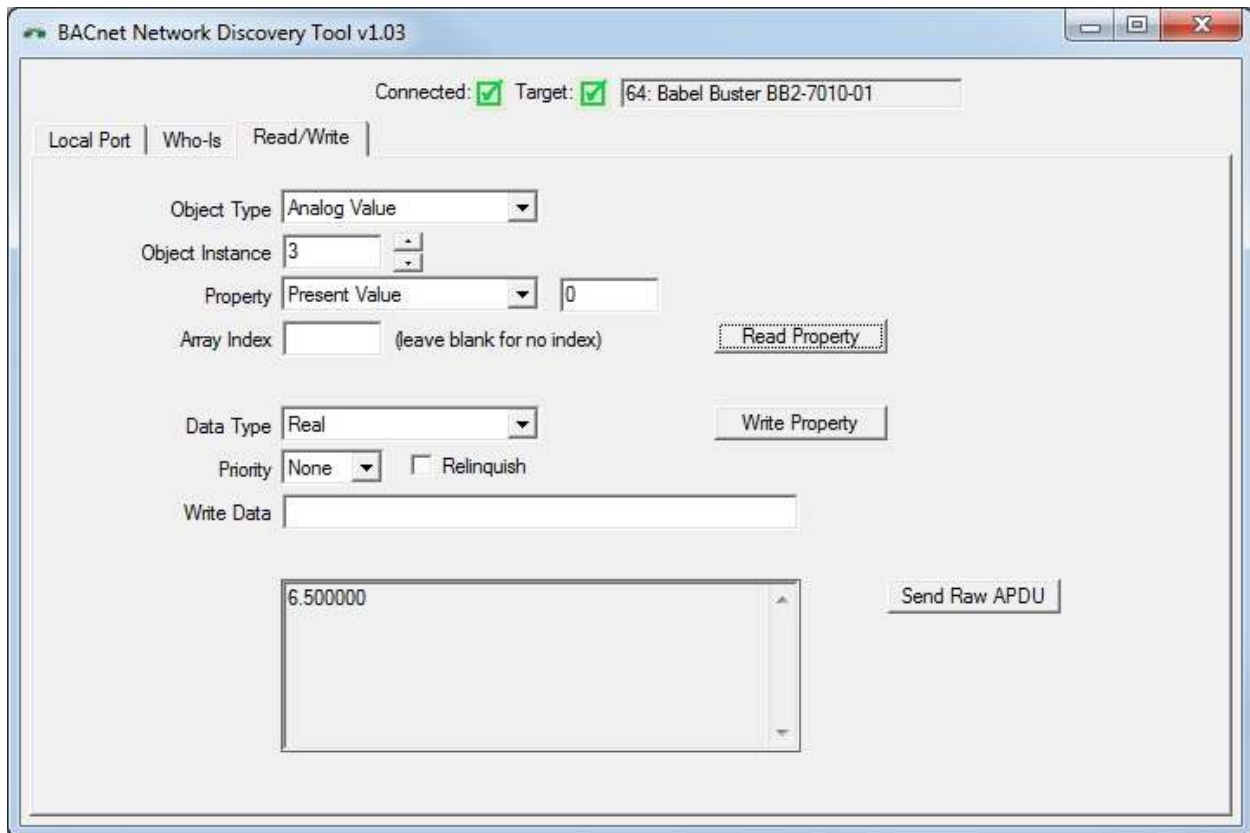
This whole process can go the other direction as well. If a Modbus RTU master was looking at the BB2-3010 as a slave, and wrote to the Modbus register corresponding to AV 4 in the BB2-3010, then the BB2-3010 would write the updated setpoint to AO 1 in the VP4-2330. Then the BB2-7010 would see the new value, and change its AV 3 accordingly, upon which ModScan would now see the new value in register 42005.



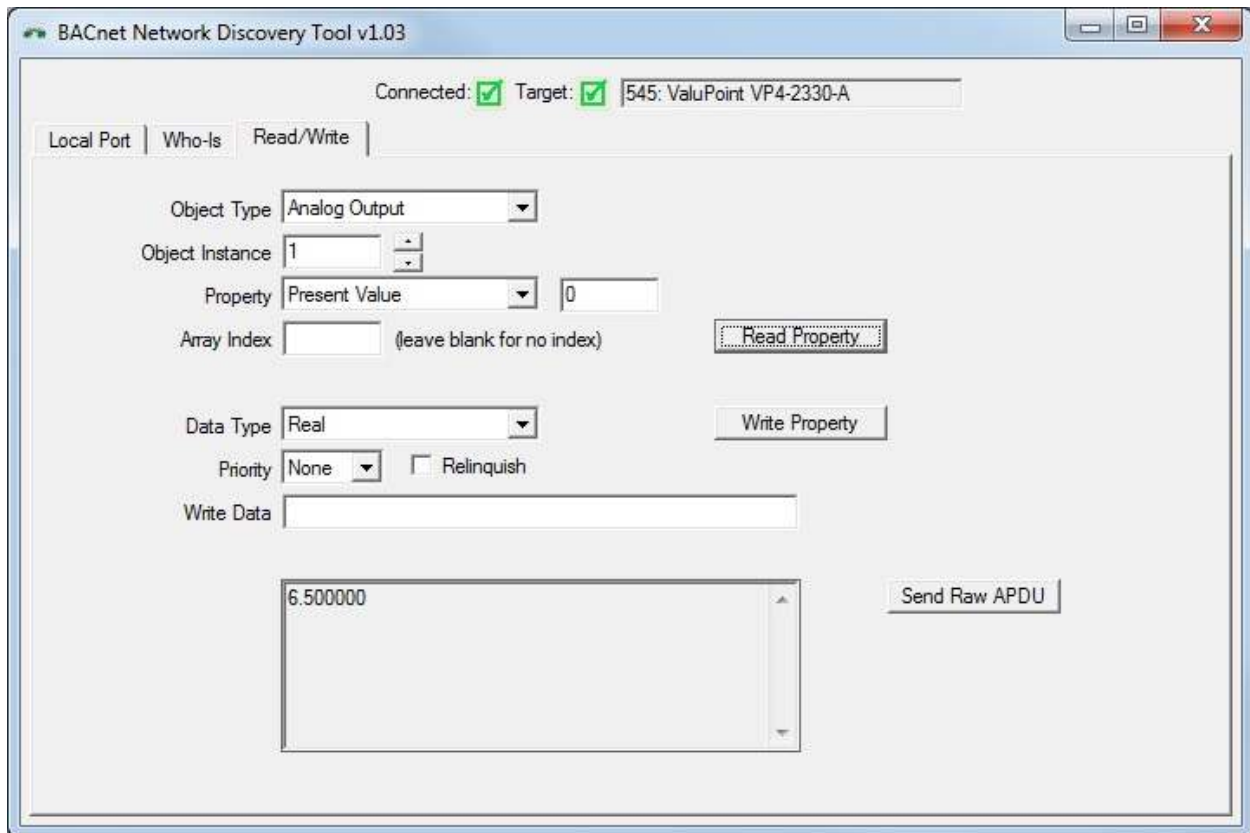
For test purposes, we can read and write AV4 in the BB2-3010 using the Network Discovery Tool.



We can also read and write AV 3 in the BB2-7010 using the Network Discovery Tool.



We can also look directly at AO 1 in the VP4-2330, and if we change AO 1 by writing directly to it, the updated value will show up in both of the AV objects noted above as well as the corresponding Modbus registers if the gateways are acting as Modbus slaves.



Article ID: 35

Created On: Mon, Nov 14, 2016 at 1:57 PM

Last Updated On: Mon, Nov 14, 2016 at 1:57 PM