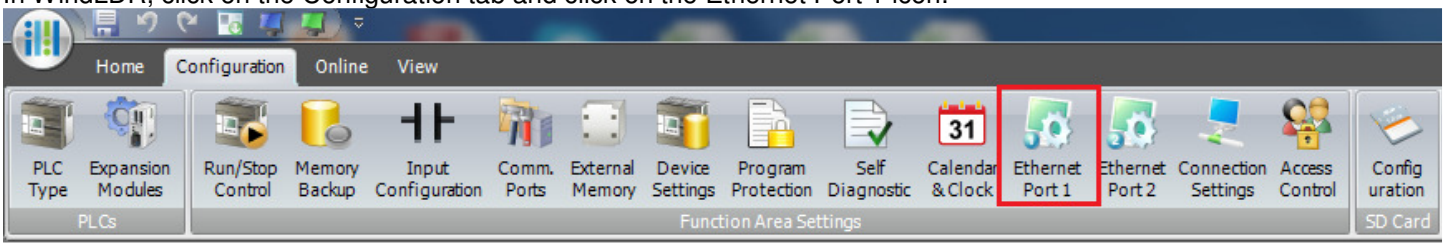
Application Note

How to Configure BACnet/IP on an FC6A Plus PLC

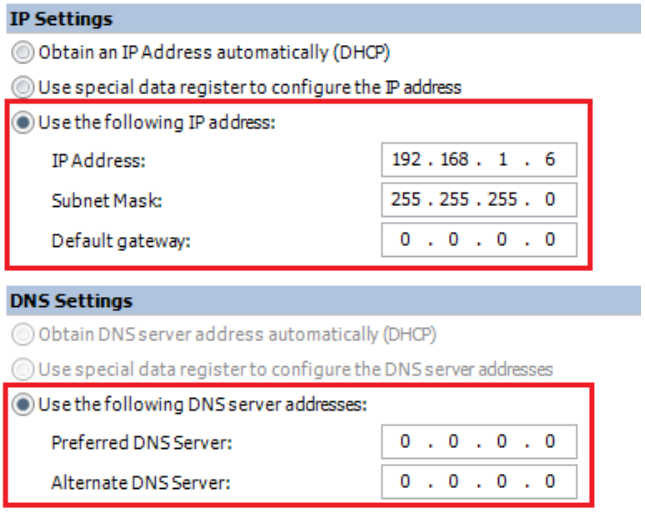
1. Configuring the Ethernet Port on the FC6A Plus and Enabling BACnet/IP

In WindLDR, click on the Configuration tab and click on the Ethernet Port 1 icon:

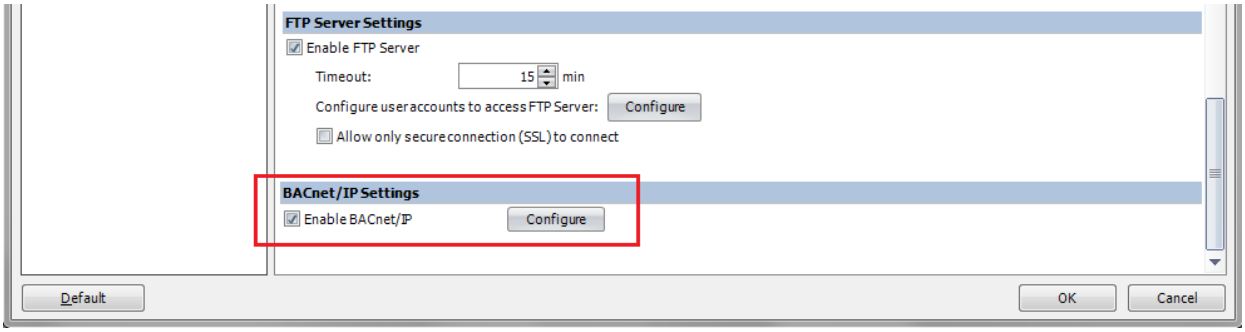


This will bring up the Function Area Settings dialog box where we can specify the IP address, subnet mask, default gateway, as well as the DNS servers.

In this example, we'll use IP Address = 192.168.1.6, Subnet Mask = 255.255.255.0, and DNS Server = 0.0.0.0



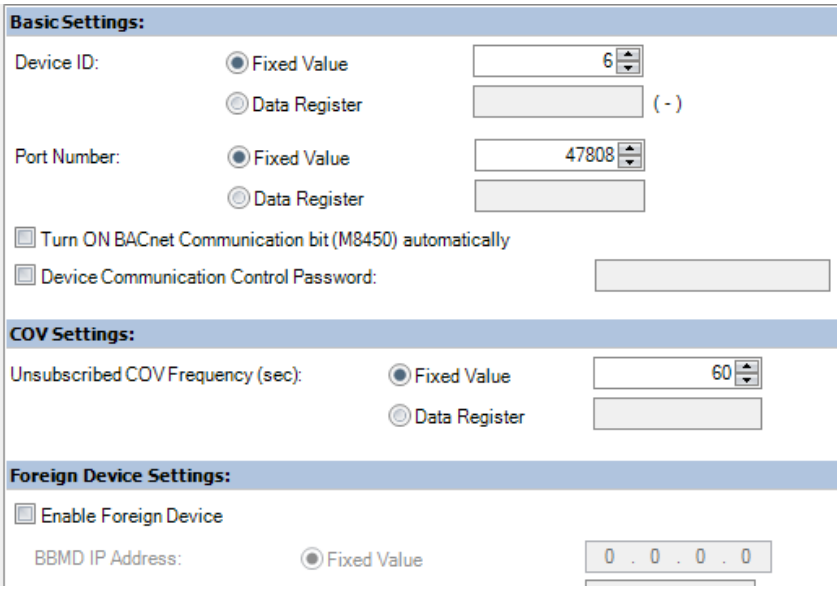
To enable BACnet/IP, **click** on the Enable BACnet/IP checkbox in the BACnet/IP Settings area and **click** Configure button.



2. BACnet/IP Settings

In this section, we'll describe the three BACnet/IP Settings dialog box:

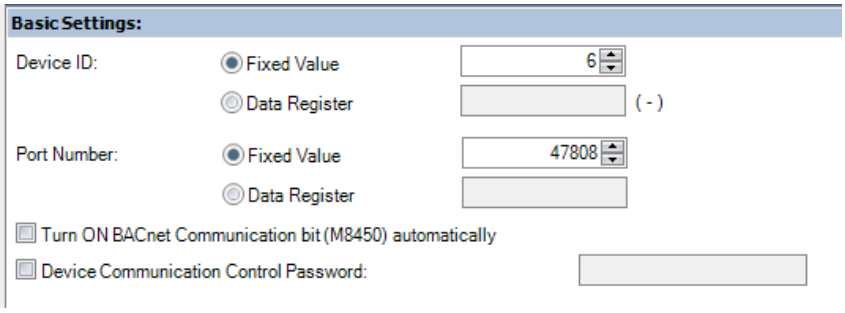
- Basic Settings
- COV (change of value) Settings
- Foreign Device Settings



Basic Settings

In the Basic Settings section, we can configure the Device ID of the PLC on the BACnet/IP network, as well as the Port Number being used. These can be hard-coded by entering a fixed value for each item, or they can be dynamic based on the value in two different data registers.

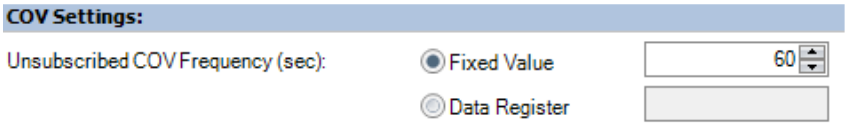
We can also choose to turn on the BACnet Communication bit (special internal relay M8450) automatically when the PLC goes into Run mode, as well as enter a password that has to be entered in order to enable device communication.



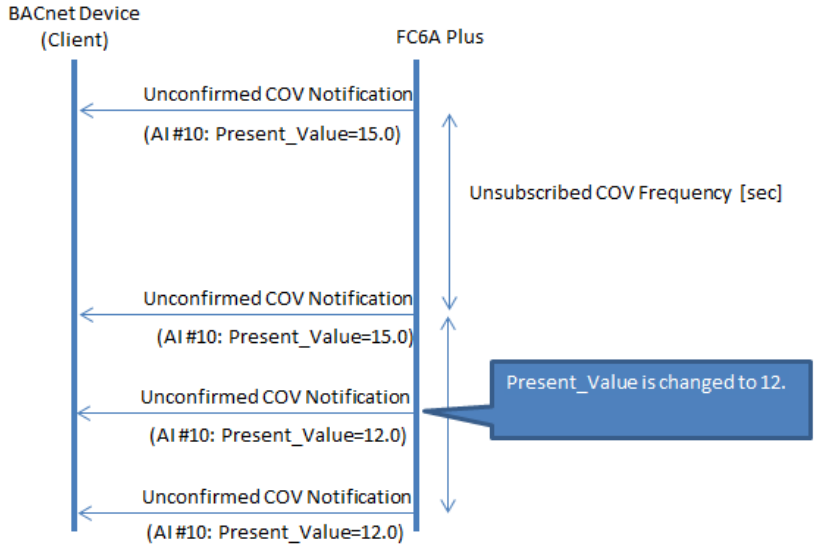
COV (change of value) Settings

In the COV Settings section, we can configure the Unsubscribed COV Frequency in seconds, either by a hard-coded fixed value or dynamically by adjusting the value in a data register.

This feature allows the FC6A Plus to send the value of unsubscribed BACnet/IP objects to the BACnet/IP client on a periodic basis for those objects that have their Send Unsubscribed COV value enabled (this will be covered later).



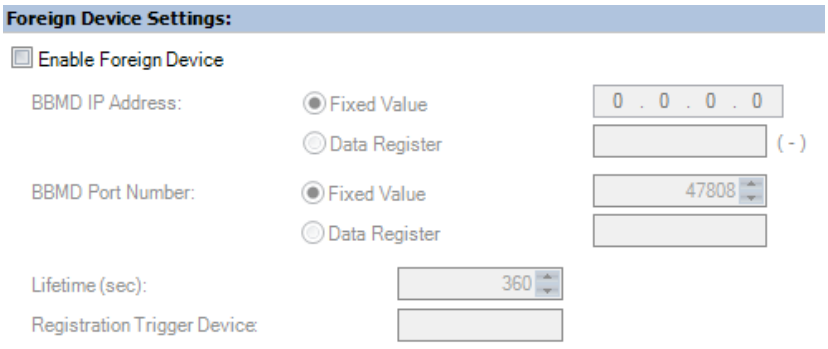
For example:



Foreign Device Settings

In the Foreign Device Settings section, we can enable the FC6A Plus as a foreign device on a BACnet/IP network. This allows a BACnet/IP Broadcast Management Device (BBMD) to receive a request from the PLC on one network and transmit it on another network.

The BBMD IP Address and Port Number can be fixed values or dynamically changed via values in data registers. The Lifetime (in seconds) is the amount of time that the FC6A Plus is registered for with the BBMD. The Registration Trigger Device is what is used to trigger the registration process.



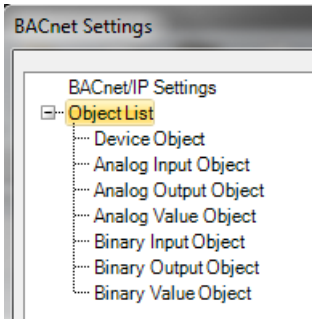
In this example, keep the Basic, COV, and Foreign Device Settings as default.

3. Object List Settings

In the BACnet/IP Settings dialog box, we can also configure the Object List. This is where we create all of the objects that will be accessible on the BACnet/IP network.

The FC6A Plus supports the following objects:

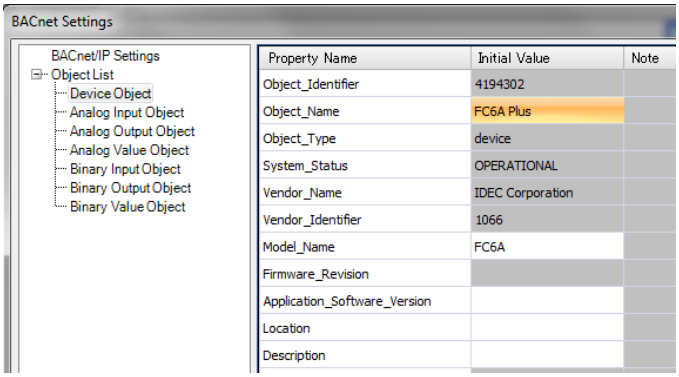
- Device Object
- Analog Input Object
- Analog Output Object
- Analog Value Object
- Binary Input Object
- Binary Output Object
- Binary Value Object



Device Object

The Device Object is for the PLC itself. We can specify the object name and the model name. We can also specify the application software version, location and description as well if we like (not required):

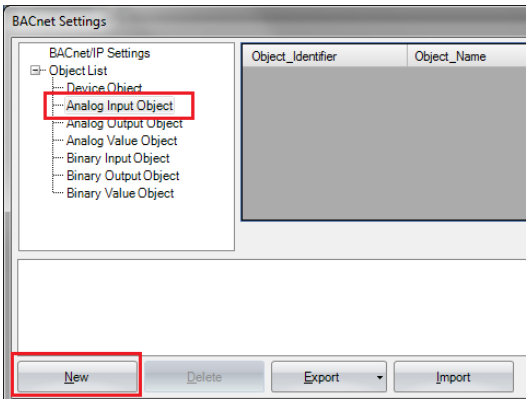
In this example, leave it as default Object_Name = FC6A Plus and Model_Name = FC6A.



Analog Input Object

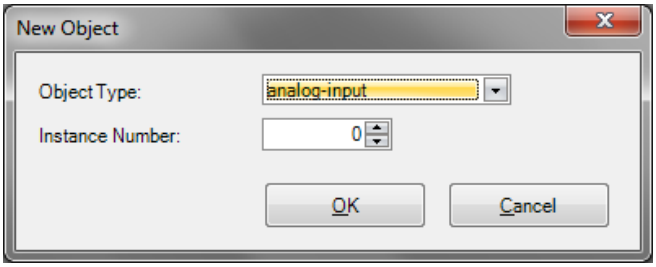
An Analog Input Object can be used to pass the value of an analog input on the FC6A Plus PLC across the BACnet/IP network.

To create a new Analog Input object, **click** on Analog Input Object in the Object List and then **click** on the New button.



This will allow us to choose the object type and specify the instance number.

In this example, set up analog input 0 as follows:



Click OK, the properties of the Analog Input object will be displayed:

Property Name	Device Control	Initial Value	Device
Object_Identifier	Disable	AI_0	Unavailable
Object_Name	Disable	Analog Input 0	Unavailable
Object_Type	Disable	analog-input	Unavailable
Present_Value	Disable	0.0	
Description	Disable		Unavailable
Device_Type	Disable		Unavailable
Status_Flags	Disable	F,F,F,F	
Event_State	Disable	Normal	Unavailable
Reliability	Disable	no-fault-detected	
Out_Of_Service	Disable	FALSE	
Units	Disable	meters-per-second-per-second	Unavailable
Resolution	Disable	1000000.0	Unavailable
COV_Increment	Disable	-1000000.0	Unavailable

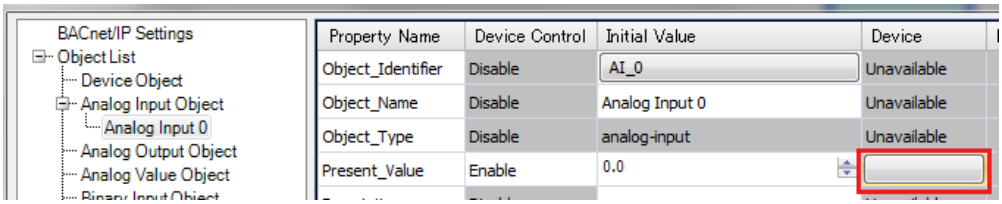
We can change the Object Name if we like.

To use a value stored in a data register, **click** and change Present_Value from Disable to **Enable**.

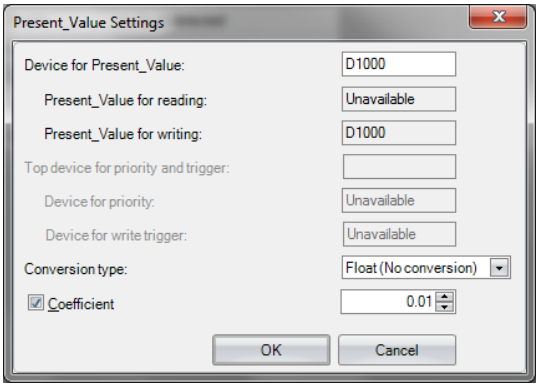
Property Name	Device Control	Initial Value	Device
Object_Identifier	Disable	AI_0	Unavailable
Object_Name	Disable	Analog Input 0	Unavailable
Object_Type	Disable	analog-input	Unavailable
Present_Value	Enable	0.0	
Description	Disable		Unavailable

This will open up a button in the Device column.

Click on the button in the Device column.



In this example, **set** Device for Present_Value as D1000 and Coefficient as 0.01.



Device for Present_Value is the data register that holds the value we want to use. Whatever we enter here is echoed in the Present_Value for writing field.

For Conversion type, we can choose the data type that we are reading – a word, integer, double word, long or float.

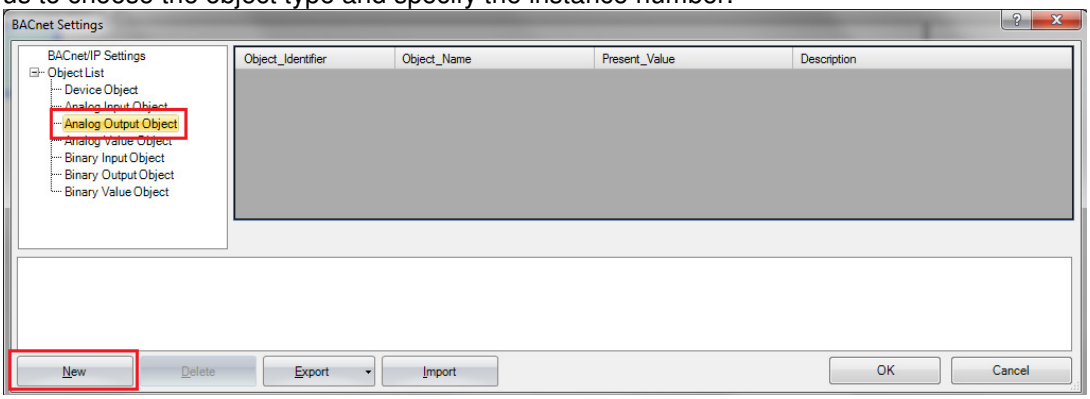
Coefficient allows you to multiply the value in the Device for Present_Value data register by a constant. If we have a value of 1000 in D1000 with a coefficient of 0.01, a value of 10.0 (1000 x 0.01) would be used:

Click OK button.

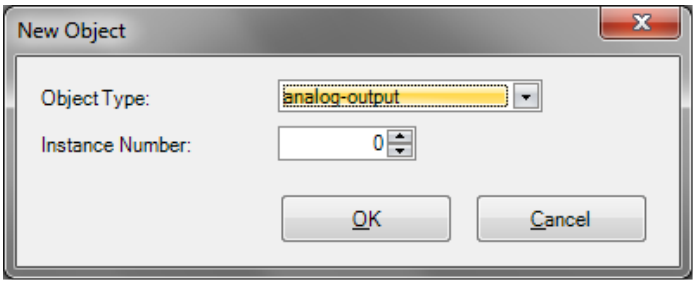
Analog Output Object

An Analog Output Object allows the FC6A Plus to take a value from the BACnet/IP client and use it to control an analog output on the PLC.

To create a new Analog Output object, **click** on Analog Output Object and then **click** on the New button. This will allow us to choose the object type and specify the instance number.



In this example, if this is analog output 0, **set** it up as follows:

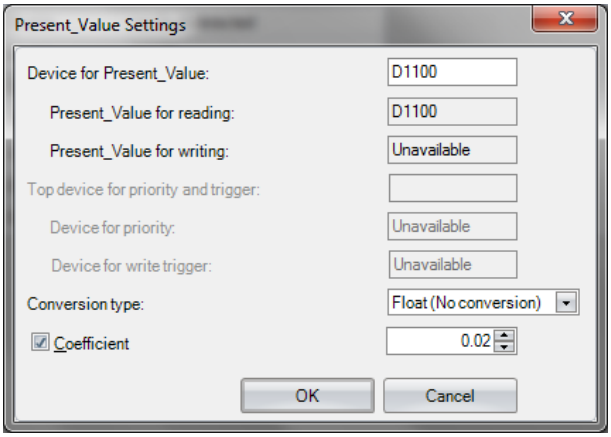


Click OK and the properties of the Analog Output object will be displayed:

Property Name	Device Control	Initial Value	Device
Object_Identifier	Disable	AO_0	Unavailable
Object_Name	Disable	Analog Output 0	Unavailable
Object_Type	Disable	analog-output	Unavailable
Present_Value	Enable	0.0	
Description	Disable		Unavailable
Device_Type	Disable		Unavailable
Status_Flags	Disable	F,F,F,F	
Event_State	Disable	Normal	Unavailable
Reliability	Disable	no-fault-detected	
Out_Of_Service	Disable	FALSE	
Units	Disable	meters-per-second-per-second	Unavailable
Resolution	Disable	1000000.0	Unavailable
Relinquish_Default	Disable	0.0	Unavailable
COV_Increment	Disable	-1000000.0	Unavailable
Priority_Array	Disable		Unavailable

To use a value stored in a data register, **click** on the button in the Device column of the Present_Value row:

In this example, **set** Device for Present_Value as D1100 and Coefficient as 0.02.



Device for Present_Value is the data register that holds the value we want to use. Whatever we enter here is echoed in the Present_Value for reading field.

For Conversion type, we can choose the data type that we are reading – a word, integer, double word, long or float.

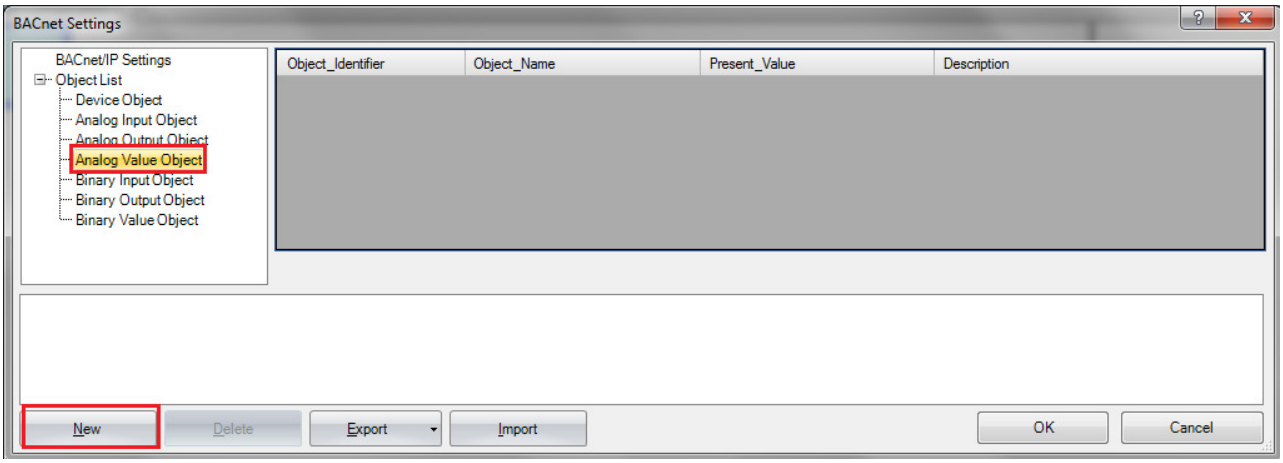
Coefficient allows you to divide the value in the Device for Present_Value data register by a constant.

For example, if we have a value of 250 in D1100 with a coefficient of 0.02, a value of 12500 (250 / 0.02) would be used:

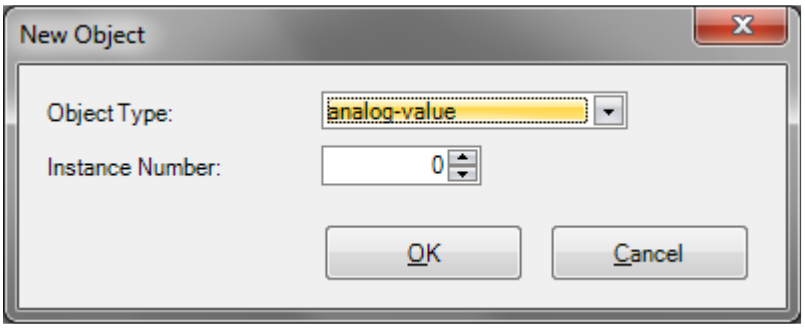
Analog Value Object

An Analog Value Object allows the BACnet/IP client to read a value from the PLC.

To create a new Analog Value object, **click** on Analog Value Object and then click on the New button. This will allow us to choose the object type and specify the instance number.



In this example, if this is analog value object, **set** it up as follows:

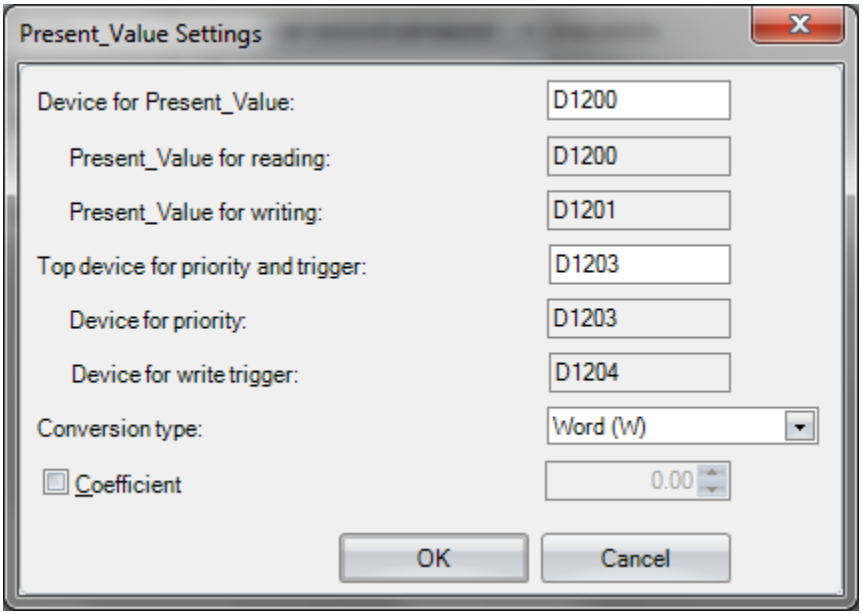


Click OK, and the properties of the Analog Value object will be displayed.

Click the button in the Device column of the Present_Value row.

Property Name	Device Control	Initial Value	Device
Object_Identifier	Disable	AV_0	Unavailable
Object_Name	Disable	Analog Value 0	Unavailable
Object_Type	Disable	analog-value	Unavailable
Present_Value	Enable	0.0	
Description	Disable		Unavailable
Status_Flags	Disable	F,F,F,F	
Event_State	Disable	Normal	Unavailable
Out_Of_Service	Disable	FALSE	
Units	Disable	meters-per-second-per-second	Unavailable
Relinquish_Default	Disable	0.0	Unavailable
COV_Increment	Disable	-1000000.0	Unavailable
Resolution	Disable	1000000.0	Unavailable
Priority_Array	Disable		Unavailable

In this example, **assign** D1200 as the Device for Present_Value, D1203 as the Top device for priority and trigger, and Conversion type = Word (W):



Device for Present_Value is the starting data register for the Present_Value for reading and Present_Value for writing registers.

Present_Value for reading is the value that is sent back to the PLC from the BACnet/IP client.
Present_Value for writing is the value that the BACnet/IP client reads from the PLC.

Top device for priority and trigger is another starting data register that will determine the addresses of Device for priority and Device for write trigger.

Device for priority is the priority level of the value in question. The higher the number, the higher the priority.

Device for write trigger is what tells the PLC to make the Present_Value for writing available to the network.

The value to be written is stored in D1201. The value that the BACnet/IP client is currently reading is stored in D1200.

The priority level of the value we are writing is stored in D1203 and the write trigger is D1204.

Let's say we move a value of 100 into D1201 and a value of 1 (highest priority) into D1203. When we're ready to make this value available, we'd write a 1 into D1204 (this will automatically be reset back to 0).

Once the value is read by the BACnet/IP client, we'll see a value of 100 in D1200.

If we want to make a new value available, we'd move that value into D1201 and write a 1 into D1204.

Note that the priority makes a difference! If we've been using a priority 1 for the values and then change the priority level to 3 for another value, that value will not be made available even if we write a 1 into D1204. A value can only be made available to the network when it has a higher priority than the last value – or equal to it, in the case that the last value made available to the network had a priority of 1.

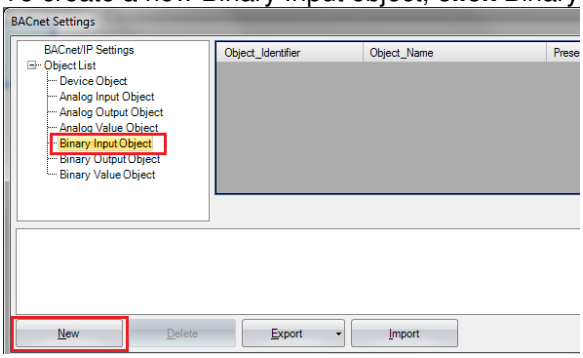
For Conversion type, we can choose the data type that we are reading – a word, integer, double word, long or float.

Coefficient allows you to multiply the value in the Device for Present_Value data register by a constant, as we've discussed earlier.

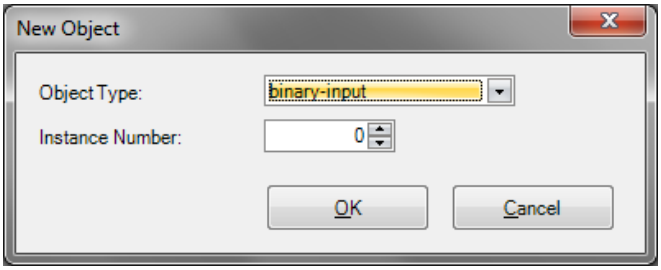
Binary Input Object

A Binary Input Object can be used to pass the value of a binary input on the FC6A Plus PLC across the BACnet/IP network.

To create a new Binary Input object, **click** Binary Input Object and **click** on the New button. :

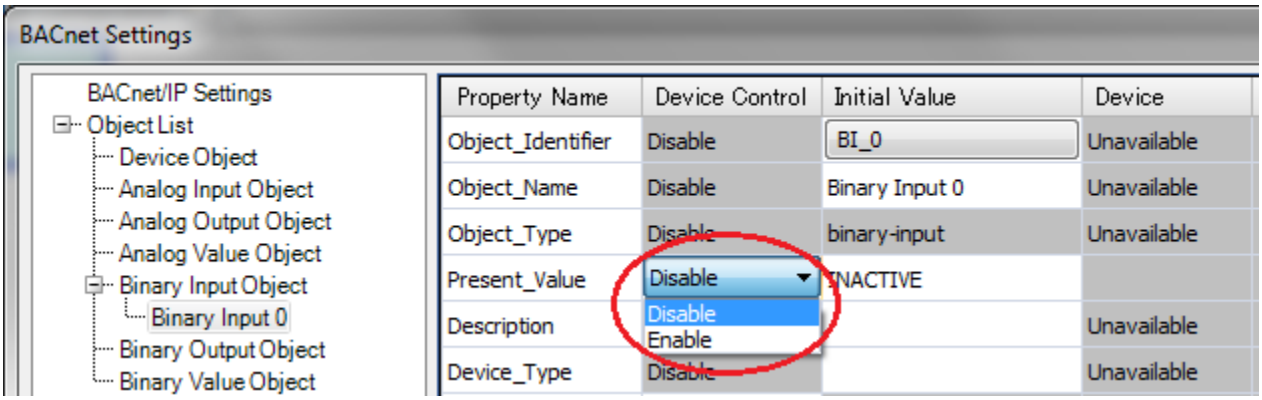


In this example, **configure** Object Type and Instance Number as follow:

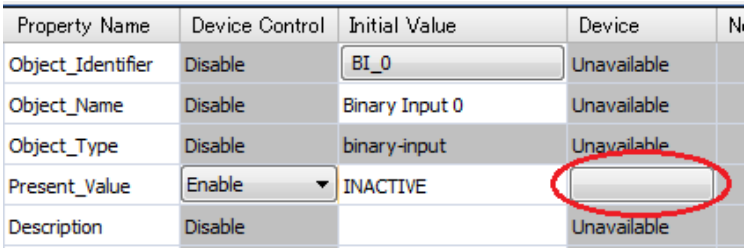


Click OK.

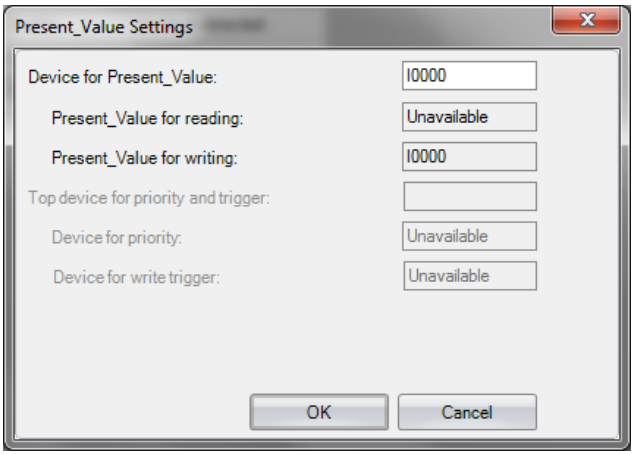
To use a value of an input, **change** Present_Value from Disable to Enable.



This will open up a button in the Device column. Click the **button** under Device column.



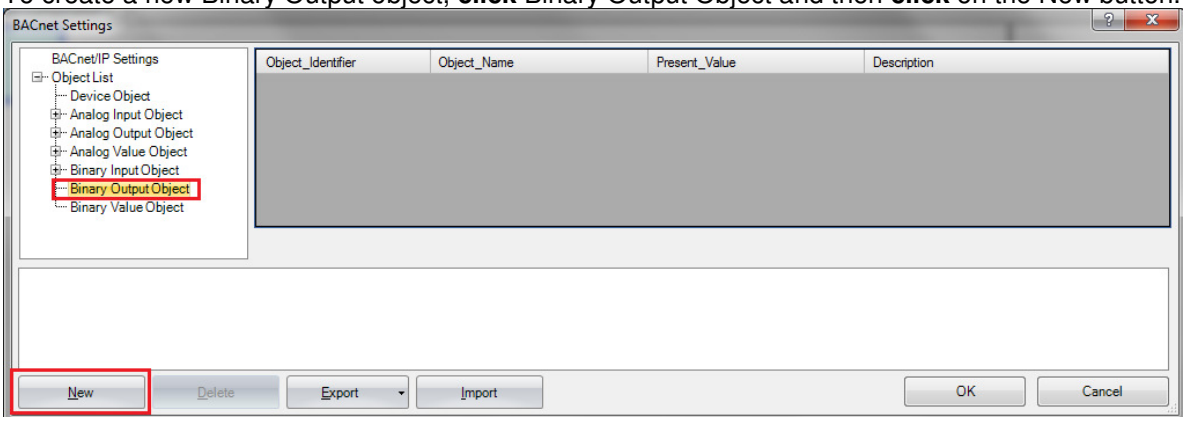
In this example, **enter** I0 for the Device for Present_Value:



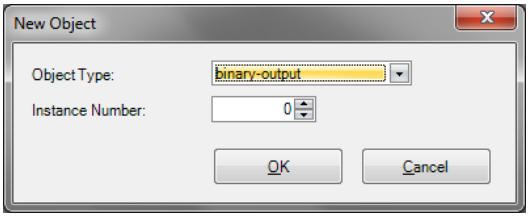
Binary Output Object

A Binary Output Object allows the FC6A Plus to take a value from the BACnet/IP client and use it to control a digital output on the PLC.

To create a new Binary Output object, **click** Binary Output Object and then **click** on the New button.

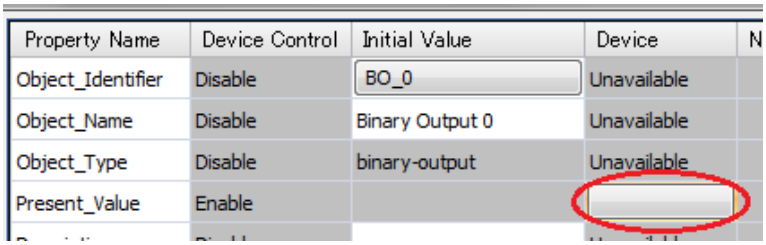


In this example, **set** it up as follows:

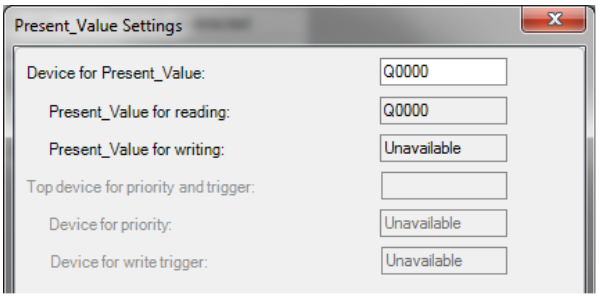


Click OK button.

Click the **button** under Device column.



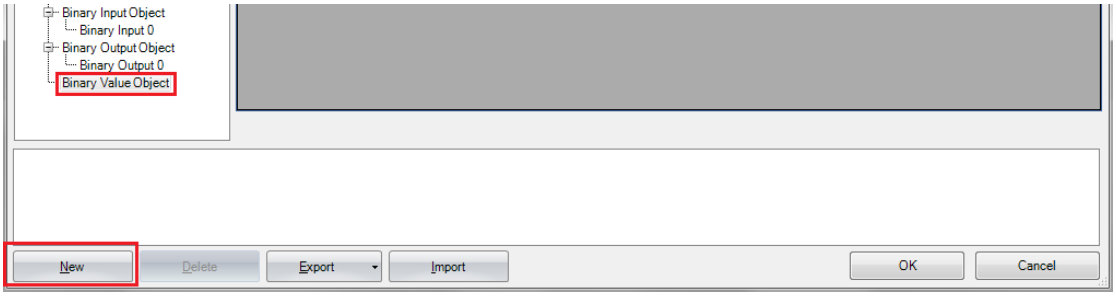
In this example, **enter** Q0 for the Device for Present_Value:



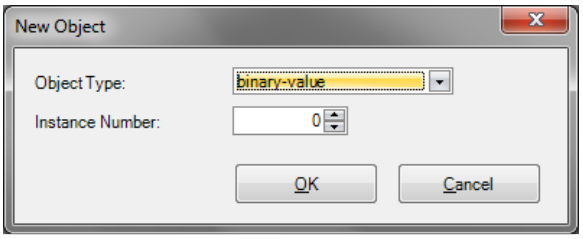
Binary Value Object

A Binary Value Object allows the BACnet/IP client to read a digital value from the PLC.

To create a new Binary Value object, **click** on Binary Value Object and then **click** on the New button.



In this example, set it up as follows:



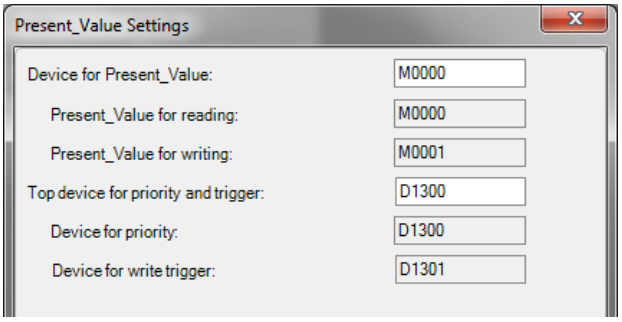
Click OK button, the properties of the Binary Value object will be displayed:

We can change the Object Name if we like.

Click on the button in the Device column of the Present_Value row.

Property Name	Device Control	Initial Value	Device
Object_Identifier	Disable	BV_0	Unavailable
Object_Name	Disable	Binary Value 0	Unavailable
Object_Type	Disable	binary-value	Unavailable
Present_Value	Enable	INACTIVE	
Description	Disable		Unavailable
Status_Flags	Disable	F,F,F,F	

In this example, **assign** M0 as the Device for Present_Value and D1300 as the Top device for priority and trigger:



Device for Present_Value is the starting internal relay for the Present_Value for reading and Present_Value for writing internal relays.

Top device for priority and trigger is the starting data register that will determine the addresses of Device for priority and Device for write trigger.

Device for priority is the priority level of the internal relay in question. The higher the number, the higher the priority. Device for write trigger is what tells the PLC to make the Present_Value for writing available to the network.

The value to be written is stored in M1. The value that the BACnet/IP client is currently reading is stored in M0.

The priority level of the value we are writing is stored in D1300 and the write trigger is D1301.

Let's say we turn on M1 and write a value of 1 (highest priority) into D1300. When we're ready to make the value of M1 available, we'd write a 1 into D1301 (this will automatically be reset back to 0).

Once the value is read by the BACnet/IP client, we'll see internal relay M0 turn on.

If we want to turn M1 off, we'd turn M1 off and then write a 1 into D1301.

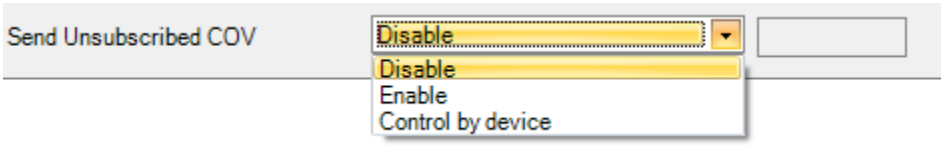
Note that the priority makes a difference! If we've been using a priority 1 for the values and then change the priority level to 3 for another value, that value will not be made available even if we write a 1 into D1301. A value can only be made available to the network when it has a higher priority than the last value – or equal to it, in the case that the last value made available to the network had a priority of 1.

Send Unsubscribed COV Object Option

You'll notice at the bottom of each object's properties, there is a drop-down that will allow you to choose how this object should be treated with regards to sending its value to the BACnet/IP client.

The BACnet/IP client can access any available object within a BACnet/IP server (the FC6A Plus) by subscribing to that object.

If the client doesn't subscribe to an object, we can decide if we want to send the value to the client anyway. This is done by the Send Unsubscribed COV drop-down. By default, this is set to Disable (the value doesn't get sent if the client doesn't request it), but we can change it to Enable (the value will get sent periodically based on the Unconfirmed COV Frequency setting that we discussed earlier) or to Control by device (the value gets sent whenever the specified device is on):



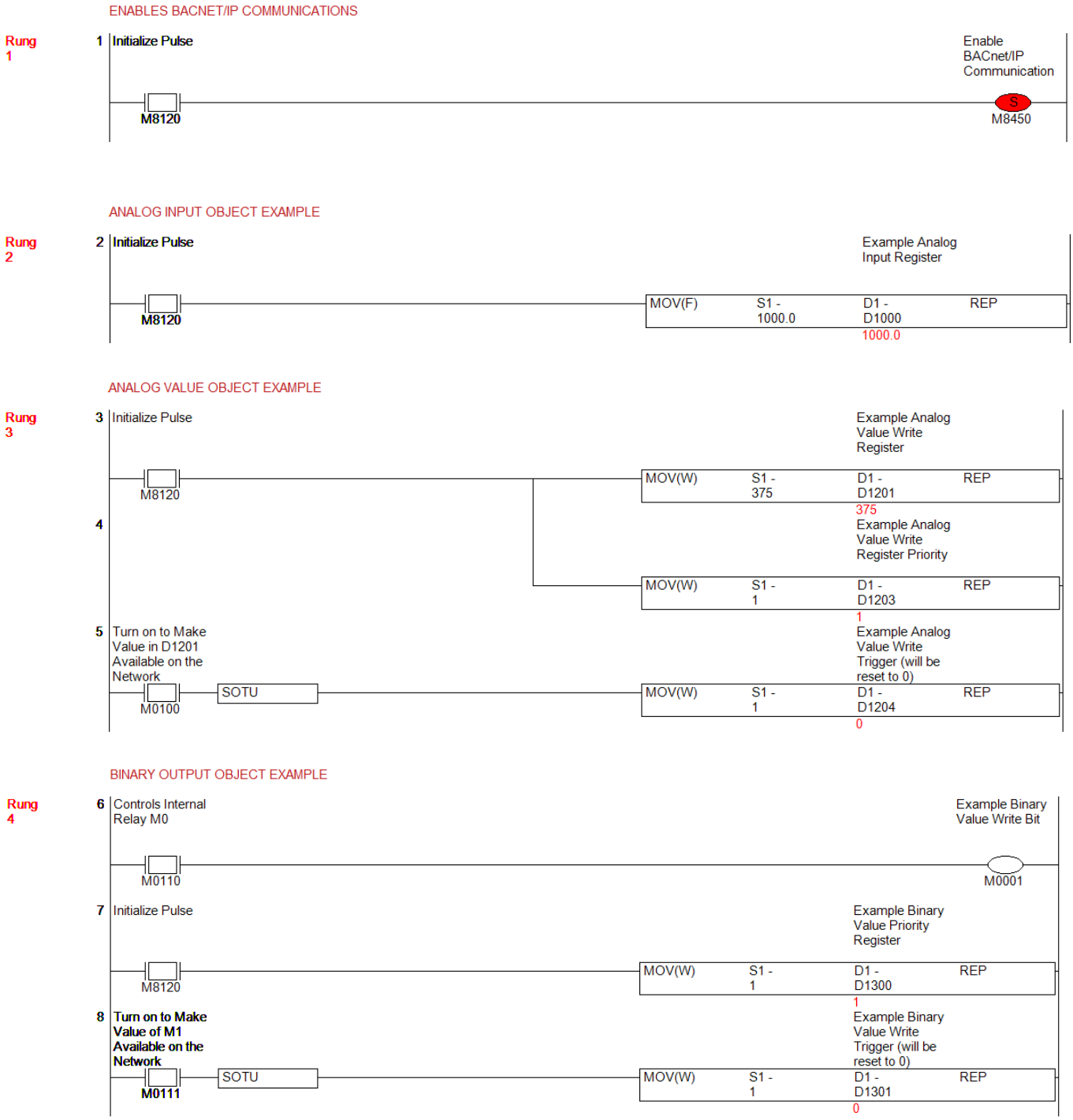
4. Special Internal Relays and Special Data Registers Used with BACnet/IP

There are two special data registers and one special internal relay used with BACnet/IP communications:

Special Data Registers		Description
D8782	BACnet status	Shows the status of BACnet/IP communication 0 : Not Executing 1 : Waiting to execute 2 : Executing 3 : Not Executing because of an error * When the status is 3, turn OFF M8450 and wait until the status changed back to 0. Once this occurs, turn ON M8450 again.
D8783	BACnet error	Shows that an error occurred in BACnet communication 0 : Normal 1 : Device ID is not correct 2 : IP address is not correct 3 : BBMD address is no correct 5 : COVU failed
Special Internal Relays		**Description**
M8450	BACnet communication	Enables / Disables BACnet Communication OFF : Disable BACnet Communication ON : Enable BACnet Communication * M8450 is turned OFF when the user program is downloaded.

5. Example PLC Program

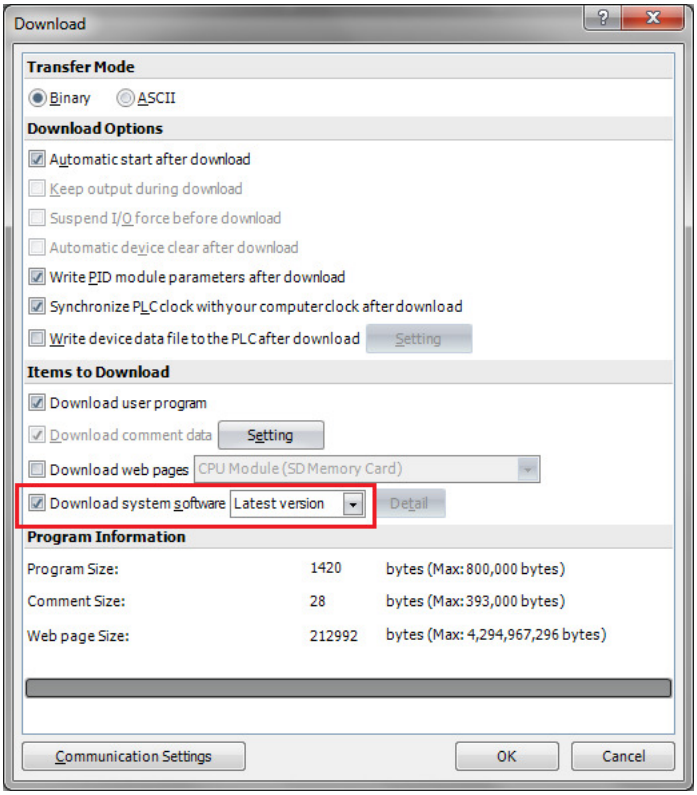
Create the ladder logic as shown below for testing.



6. Downloading to the FC6A Plus

When you download your program to the FC6A Plus PLC for the first time, please make sure you select the Download system software check box and choose Latest version in order to download the latest version of the firmware that supports BACnet/IP to the PLC:

Click OK button to download.



7. BACnet/IP Client Software

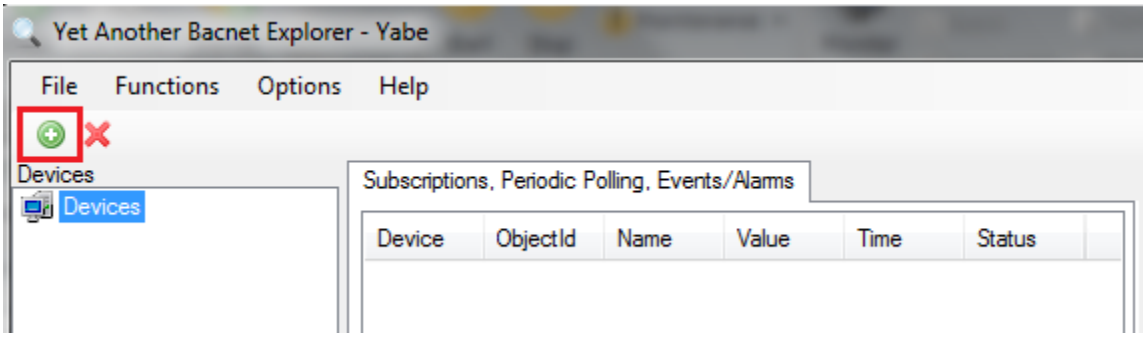
The FC6A Plus is designed to connect as a BACnet/IP server to a BACnet/IP client. If you don't have a BACnet/IP client that you are currently using, there are several freeware BACnet/IP client software packages available online.

One that was used in testing of the example PLC program mentioned earlier in the document is called Yet Another BACnet Explorer (YABE). It can be downloaded via this link: https://sourceforge.net/projects/yetanotherbacnetexplorer/. Please note that YABE is not an IDEC software package, nor is it endorsed by IDEC. If you choose to download and/or install this software, you do so at your own risk.

This section shows how to set up the YABE software to communicate with an FC6A PLC over BACnet/IP. It also shows how to test communications of each object we created using the example PLC program mentioned in section 5.

Setting Up the Connection

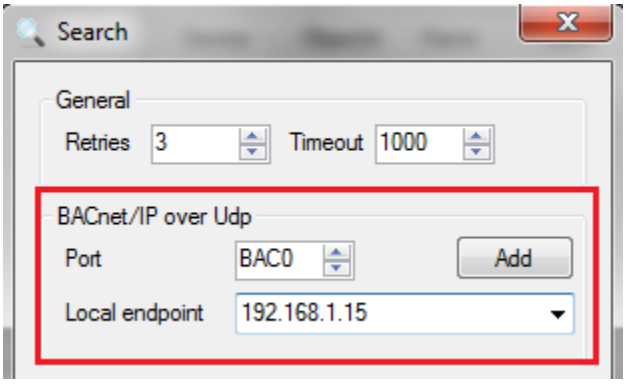
To set up a new device in YABE, click on the + sign:



This will bring up the Search dialog box. In the BACnet/IP over Udp section, set Port to BAC0 and Local endpoint to the IP address of the Ethernet port on your computer that you are using to connect to Ethernet Port 2 on the FC6A Plus.

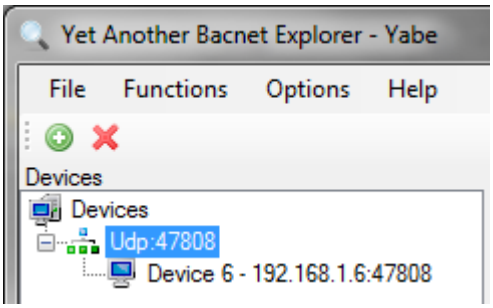
In this example, the IP address of Ethernet Port 1 on the FC6A Plus PLC is 192.168.1.6. In order for our computer's Ethernet port to be in the same network as the PLC, it also needs to have an IP address of 192.168.1.x, with a subnet mask of 255.255.255.0.

We've set the IP address of the computer's Ethernet port to 192.168.1.15, so that is what we will enter for Local endpoint:



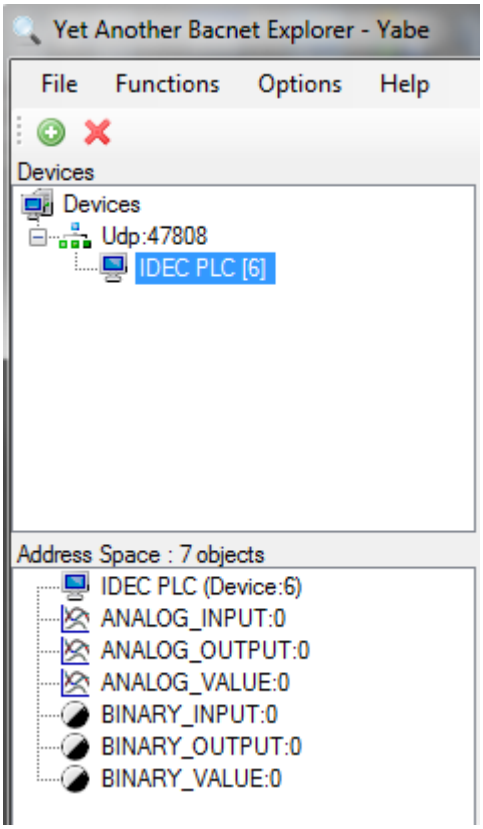
Click the Add button to search for BACnet/IP servers that are connected to the same network as 192.168.1.15.

If the FC6A Plus is found, it will appear in the list:



Clicking on the PLC – shown as Device 6 – 192.168.1.6:47808 – will show all of the BACnet/IP objects that we created.

You'll also notice that the object name that we assigned in the Device Object (IDEC PLC) now shows as well:



If the FC6A Plus PLC is not found, we can check the communication status of the network.

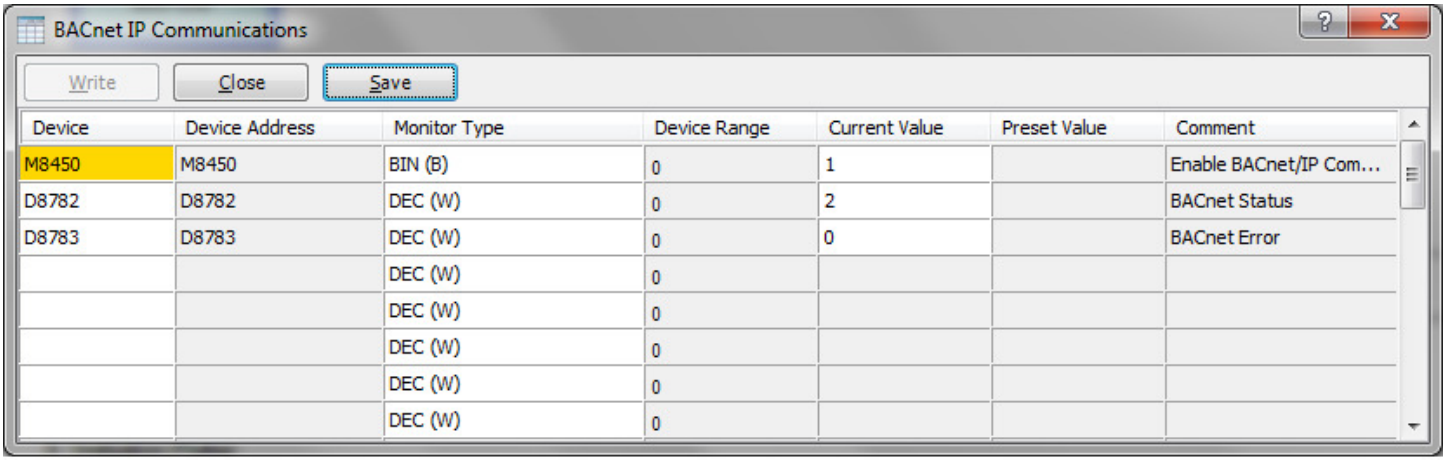
Checking Communications

If the PLC isn't showing up in YABE, check to make sure that the Ethernet port on your computer is in the same network as the PLC as mentioned earlier.

Check to make sure you are connected to Ethernet Port 1 on the FC6A Plus CPU. Ethernet Port 1 is the "top" Ethernet port.

We can also check the PLC's communication status by looking at the special internal relays and special data registers used for BACnet/IP.

In WindLDR software, go into Monitor mode, select the Custom drop-down and choose custom monitor box and enter M8450, D8782 and D8783.



Device	Device Address	Monitor Type	Device Range	Current Value	Preset Value	Comment
M8450	M8450	BIN (B)	0	1		Enable BACnet/IP Com...
D8782	D8782	DEC (W)	0	2		BACnet Status
D8783	D8783	DEC (W)	0	0		BACnet Error
		DEC (W)	0			
		DEC (W)	0			
		DEC (W)	0			
		DEC (W)	0			
		DEC (W)	0			

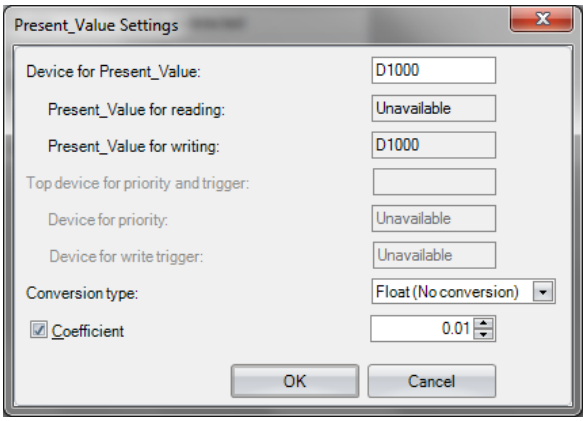
- M8450 is on, which means the PLC has been enabled for BACnet/IP communications.
- D8782 is 2, which means the communication is executing.
- D8783 is 0, which means there are no errors.

If M8450 is off, please turn it on. If you have numbers other than 2 in D8782 and 0 in D8783, please refer to section 4 of this document, which outlines the various error codes for each of these special data registers.

In rung 1 of the PLC ladder program, we set M8450 to be on at the first scan of the PLC:

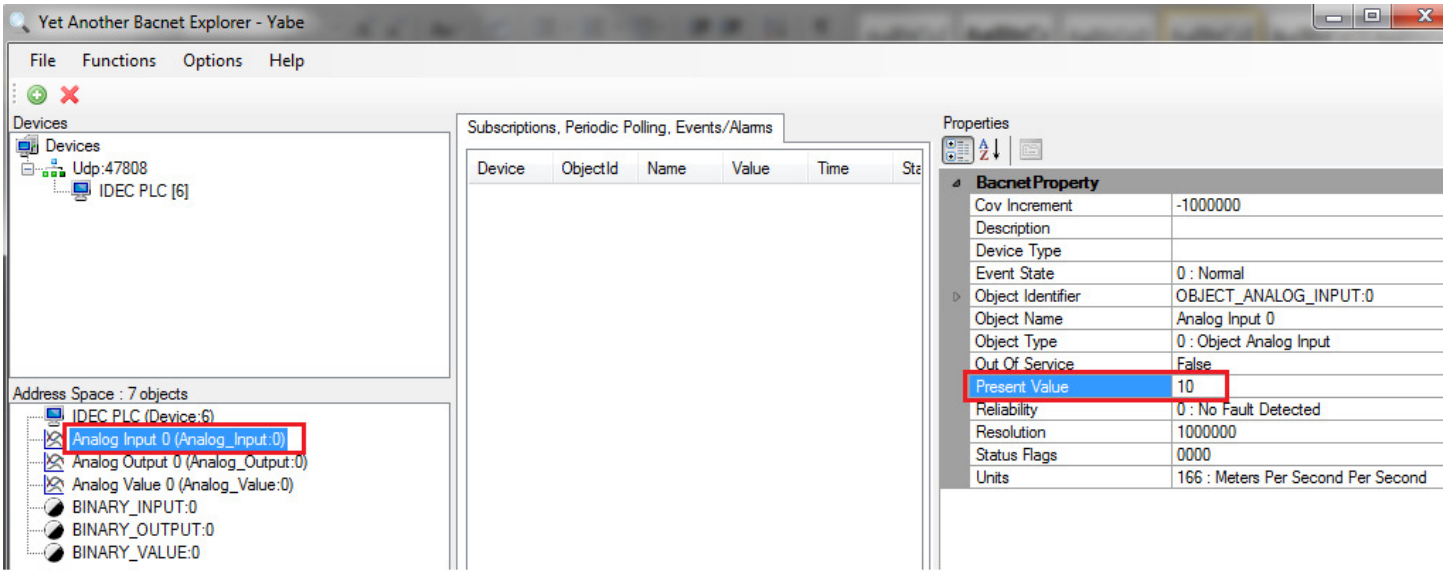
In rung 2 of the PLC ladder program, we move a float value of 1000.0 to data register D1000.

When we created the Analog Input object earlier, we set it up to use data register D1000 as a float, with a coefficient of 0.01:



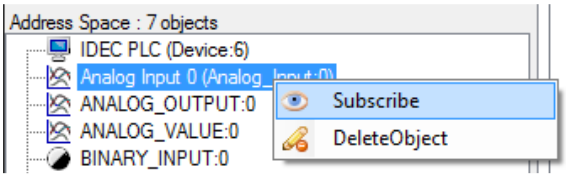
This means that the value we are providing over BACnet is the value in D1000 multiplied by 0.01. In our case, we should see a value in YABE for the Analog Input object of 1000 x 0.01 = 10.

In YABE, if we click on the Analog Input object, we should see the value of 10:

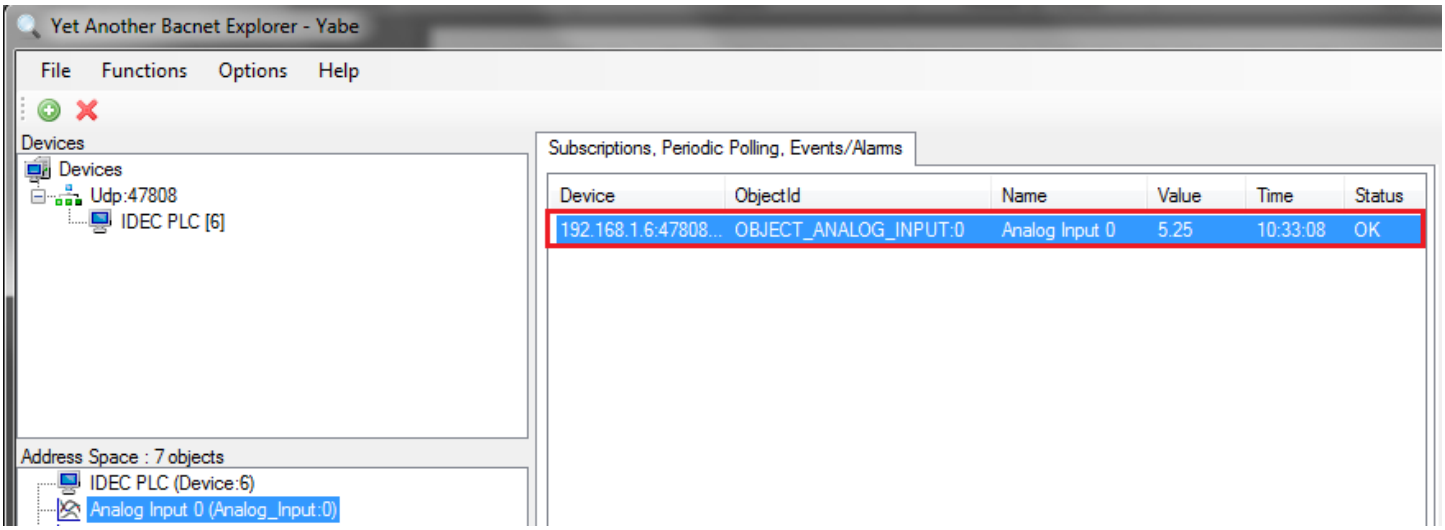


If the Client wants to regularly get updates from an object, it can subscribe to an object. This way, if the value of the object changes, this change will be shown in the client.

In YABE, you can subscribe to an object by right-clicking on the object and selecting Subscribe:

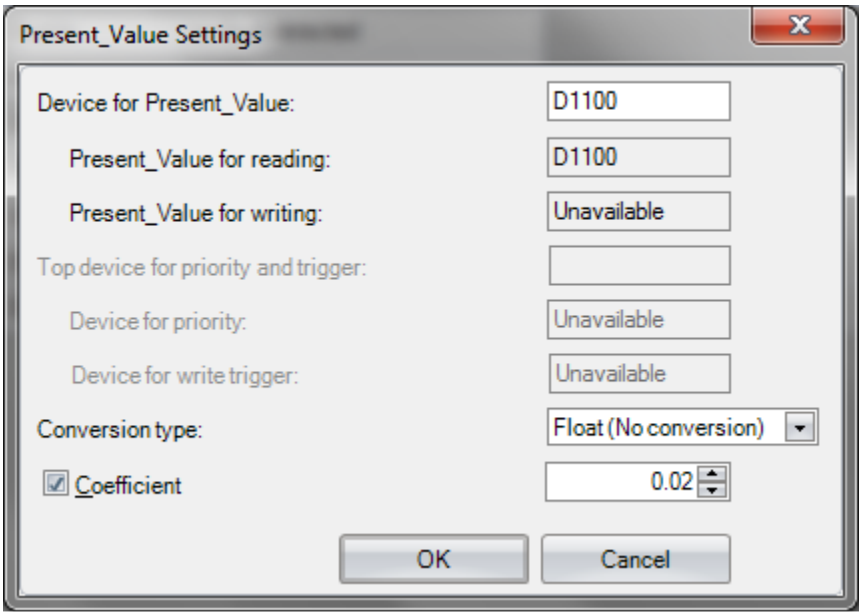


If we subscribe to the Analog Input object and then change the value in D1000 in the PLC to 525, we should see the value in YABE for the Analog Input update to 525 x 0.01 = 5.25:

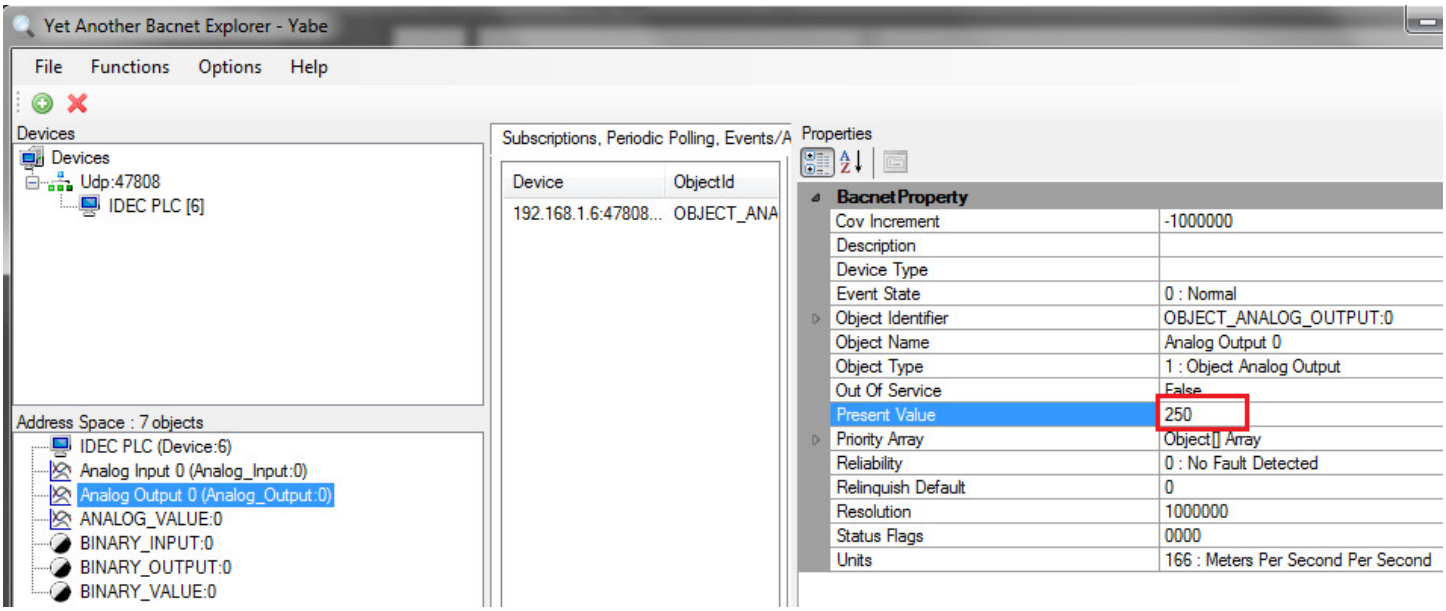


Testing the Analog Output Object

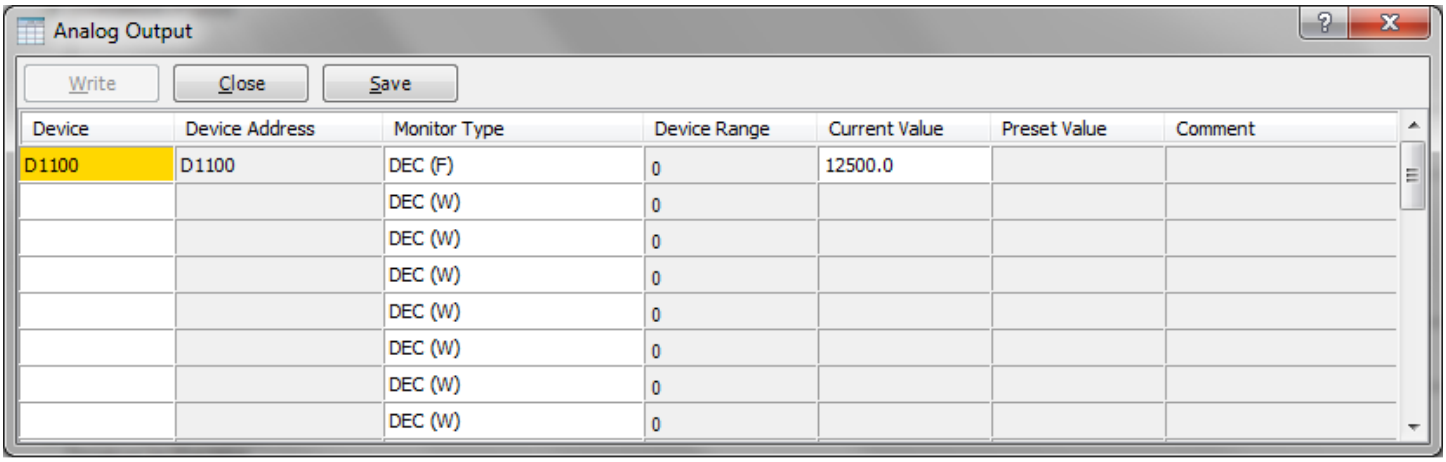
When we created our Analog Output object, we used data register D1100 as a float with a coefficient of 2. This means that whatever value the BACnet/IP Client is sending to this object should be multiplied by 0.02, with that result shown in D1100:



In YABE, we can force the value of the Analog Output object by clicking on the object to select it, then clicking in the white space beside the Present Value column, entering a value and pressing the ENTER key:

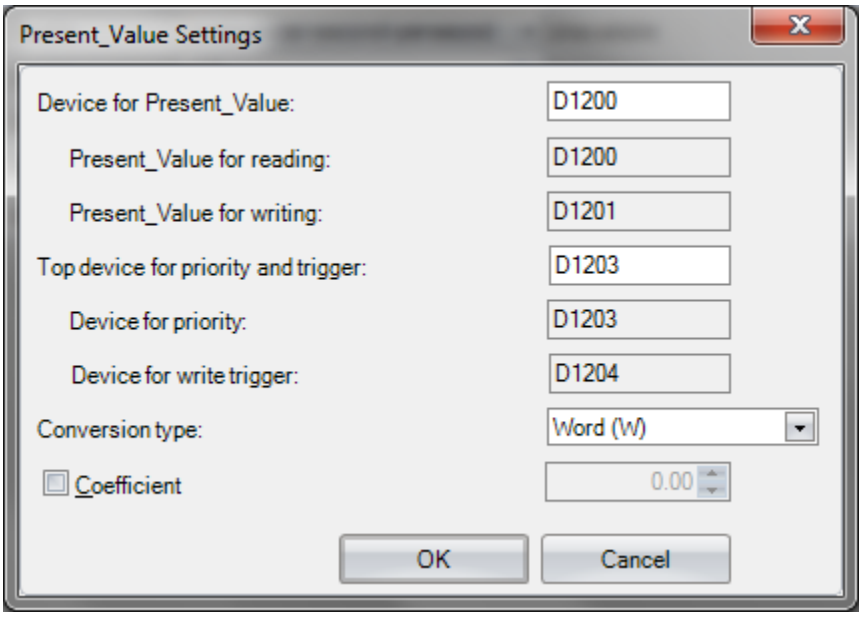


If you open the Analog Output custom monitor dialog box in WindLDR, we should have a value of 250 / .02 = 12,500:

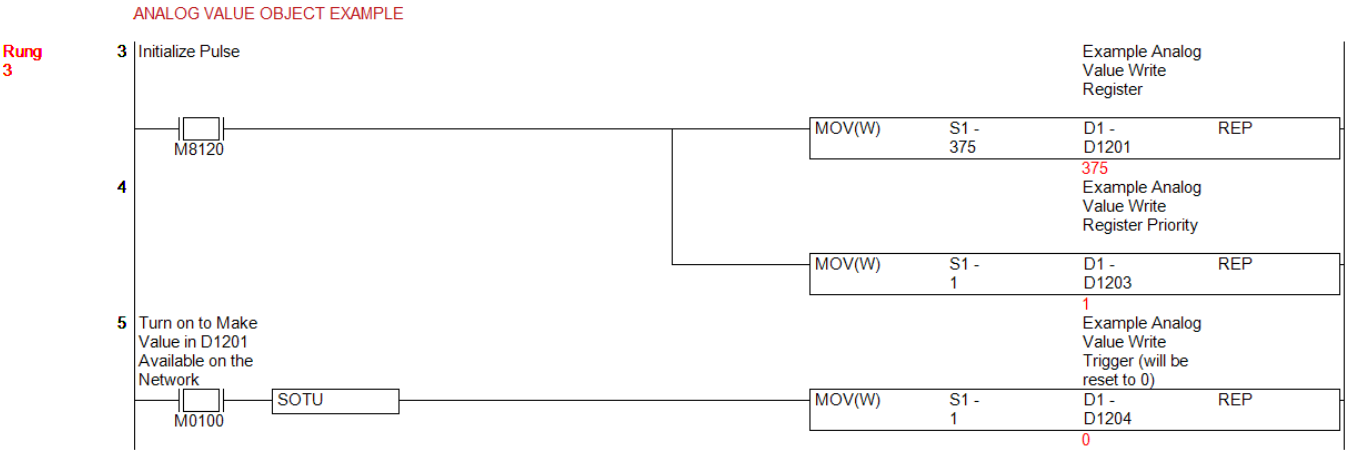


Testing the Analog Value Object

When we created the Analog Value object, we used the following data registers:

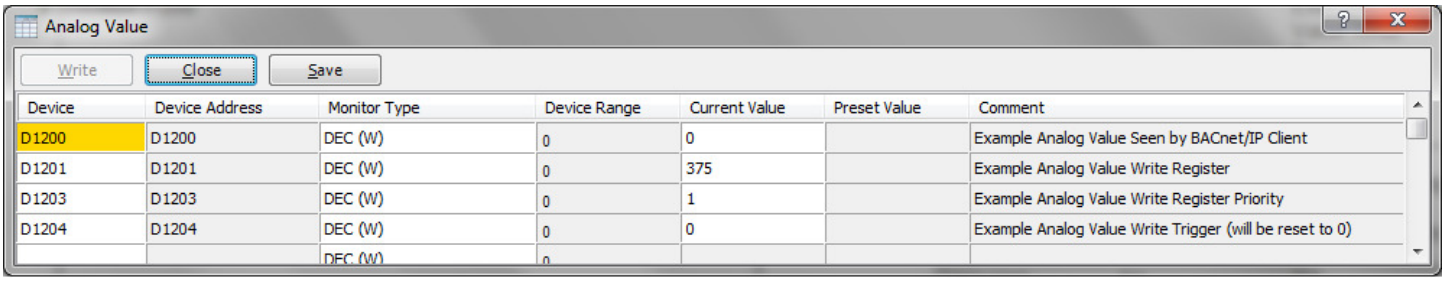


In the PLC ladder program, we have the following logic:

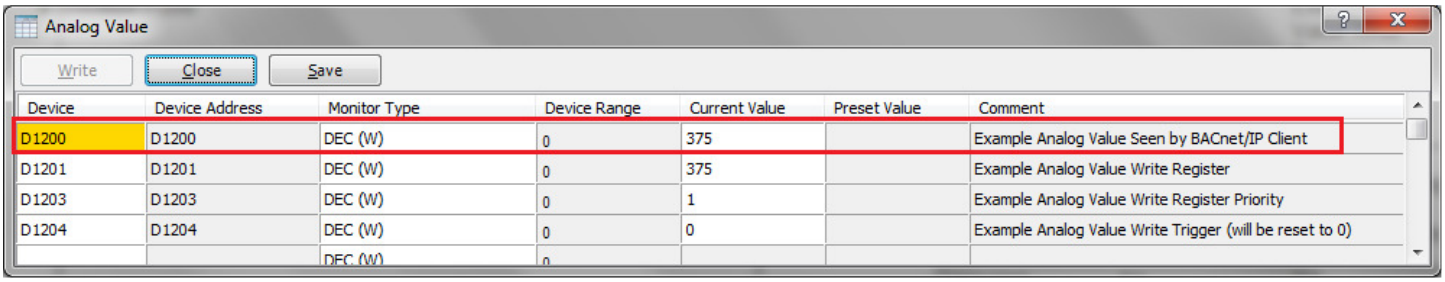


When we turn on M100, the value in D1201 will be made available to the BACnet/IP client. The value that the client is reading gets reported in D1200.

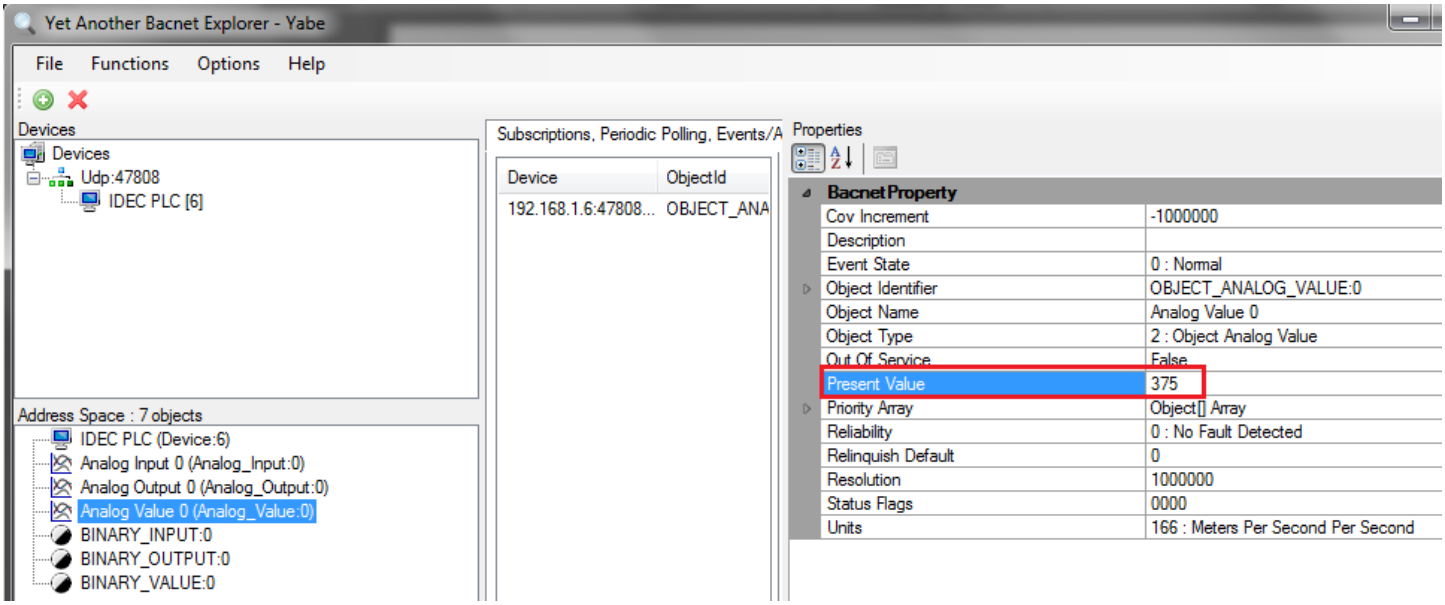
If you open the Analog Value custom monitor box, you'll see that we currently have a value of 0 in D1200 and 375 in D1201:



If we turn on M100 in the logic and then turn it back off, that will make the value in D1201 available. We should see D1200 change to 375:

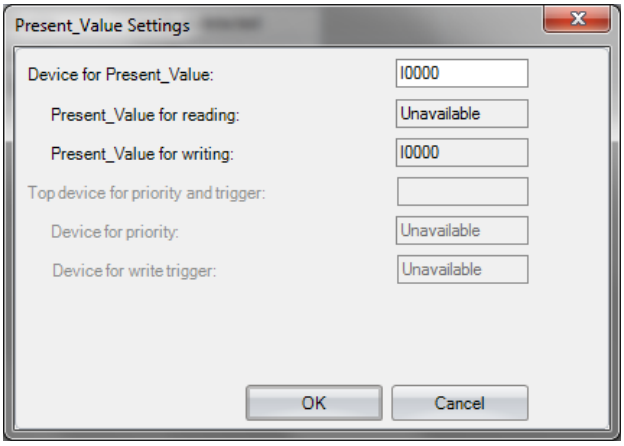


We can also confirm this in YABE by clicking on the Analog Value object and looking at the Present Value:

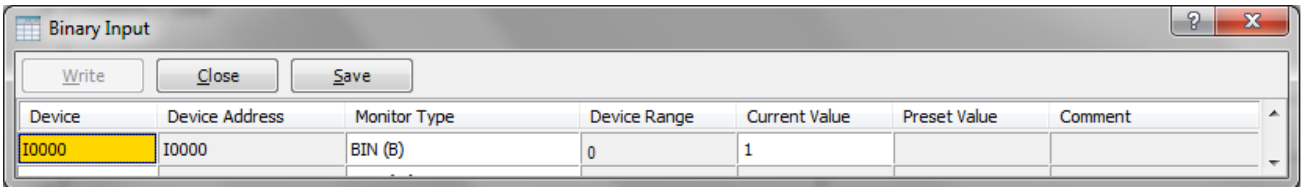


Testing the Binary Input Object

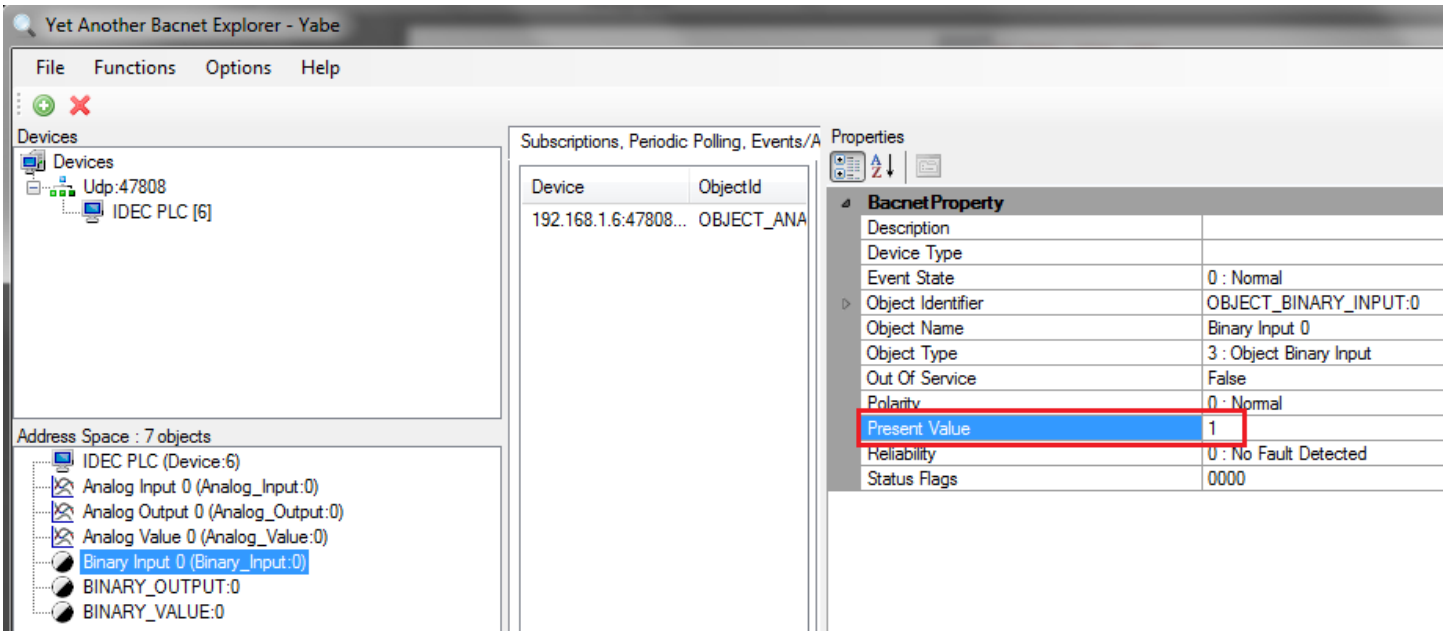
When we created the Binary Input object, we used input I0:



If you open the Binary Input custom monitor dialog box, you'll see the status of I0. In this case, I0 in the PLC is on:

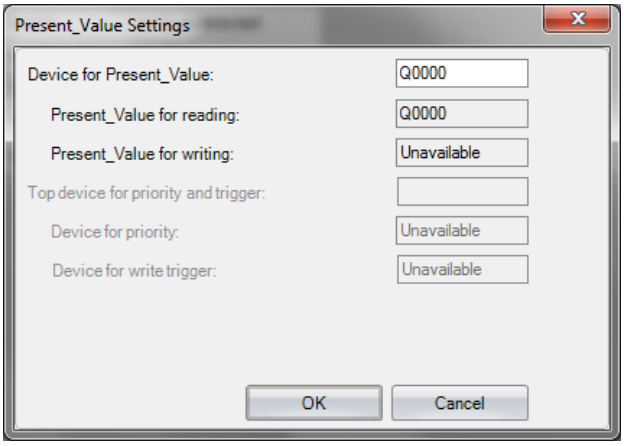


We can also confirm this in YABE by clicking on the Binary Input object and looking at the Present Value:

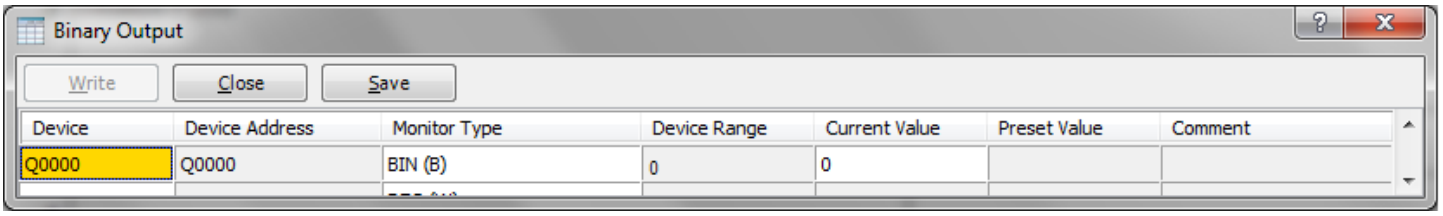


Testing the Binary Output Object

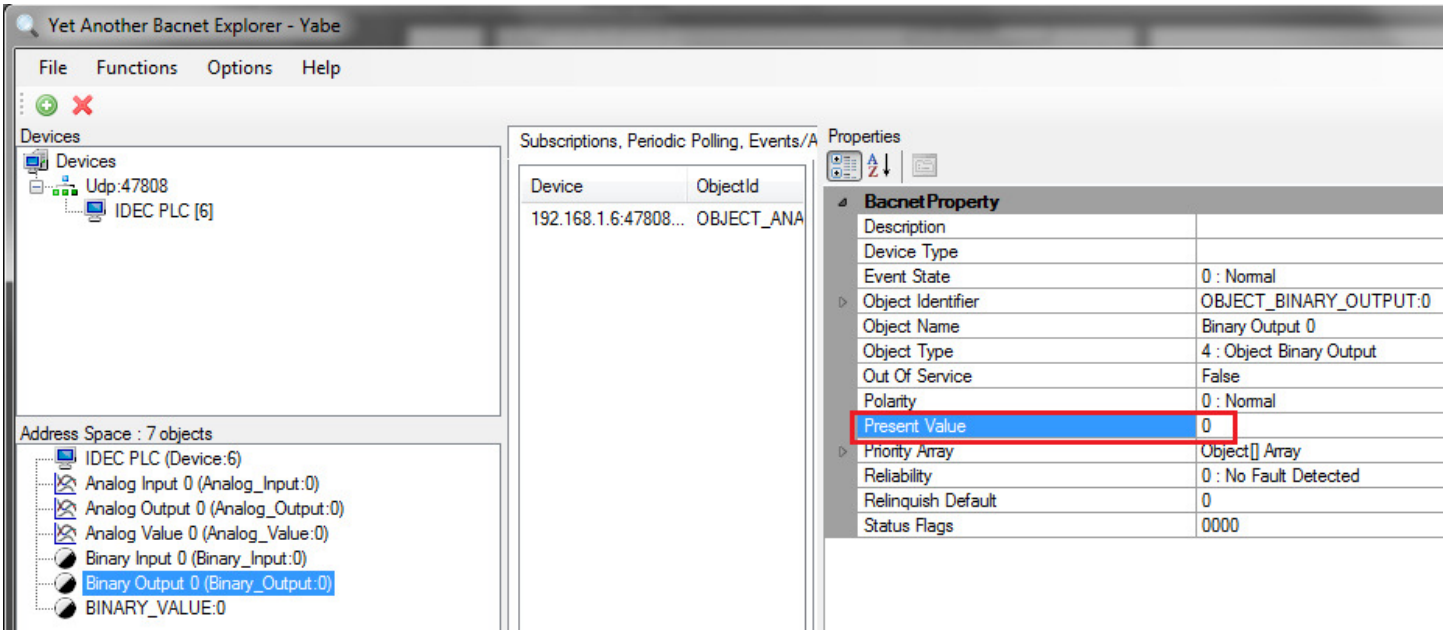
When we created the Binary Output object, we used Q0:



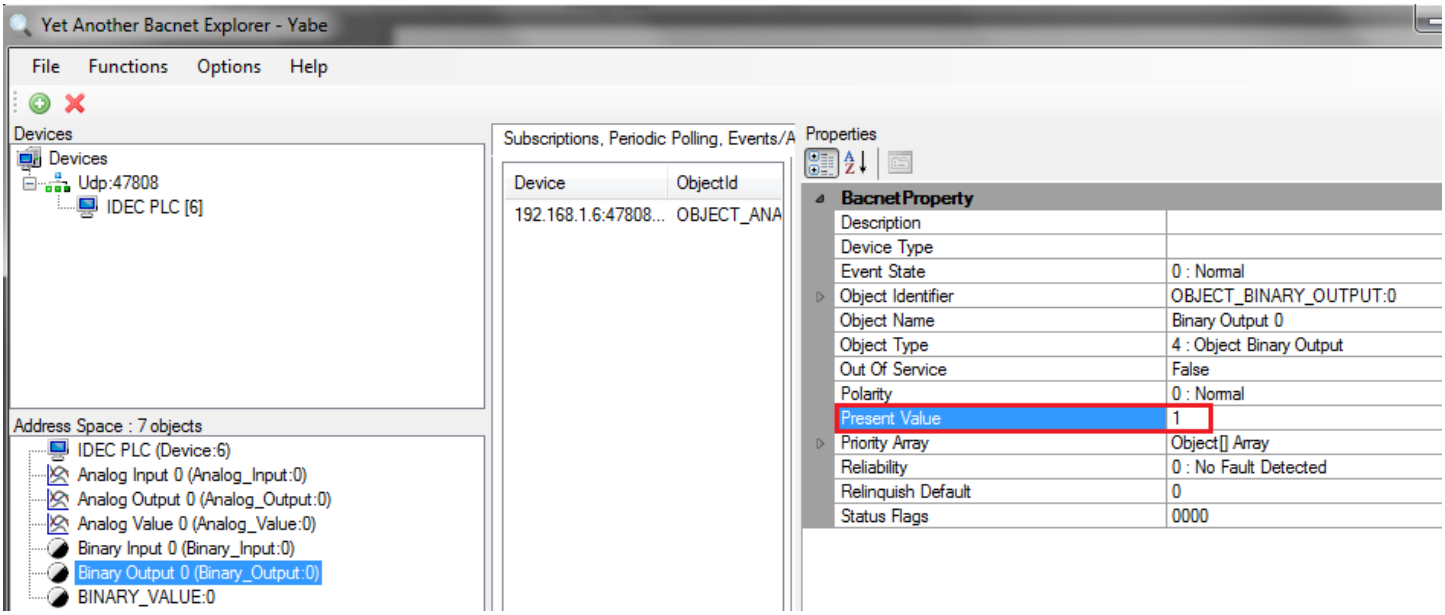
If you open the Binary Output custom monitor dialog box, output Q0 on the PLC is currently off:



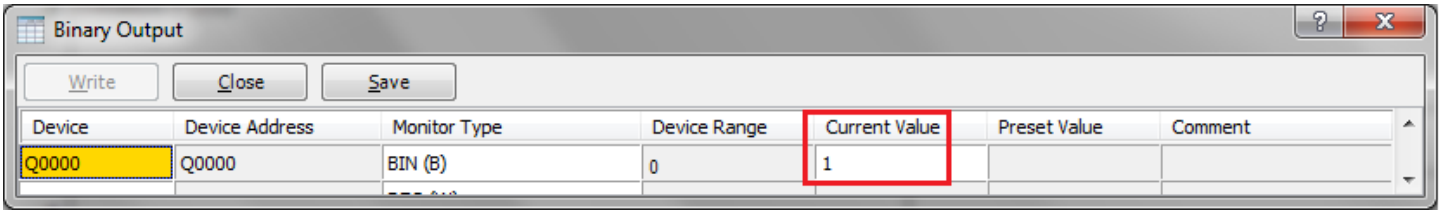
We can confirm this in YABE as well by clicking on the Binary Output object and looking at its Present Value:



If we change the Present Value to 1:

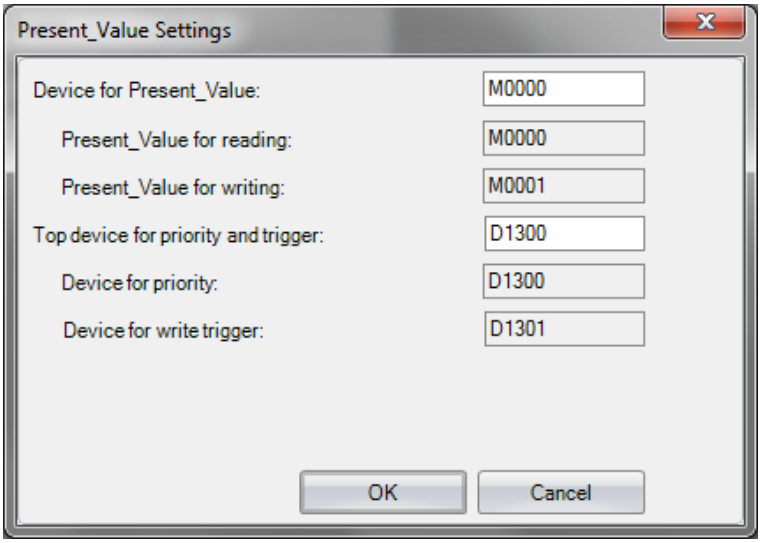


We should also see this update in the PLC:

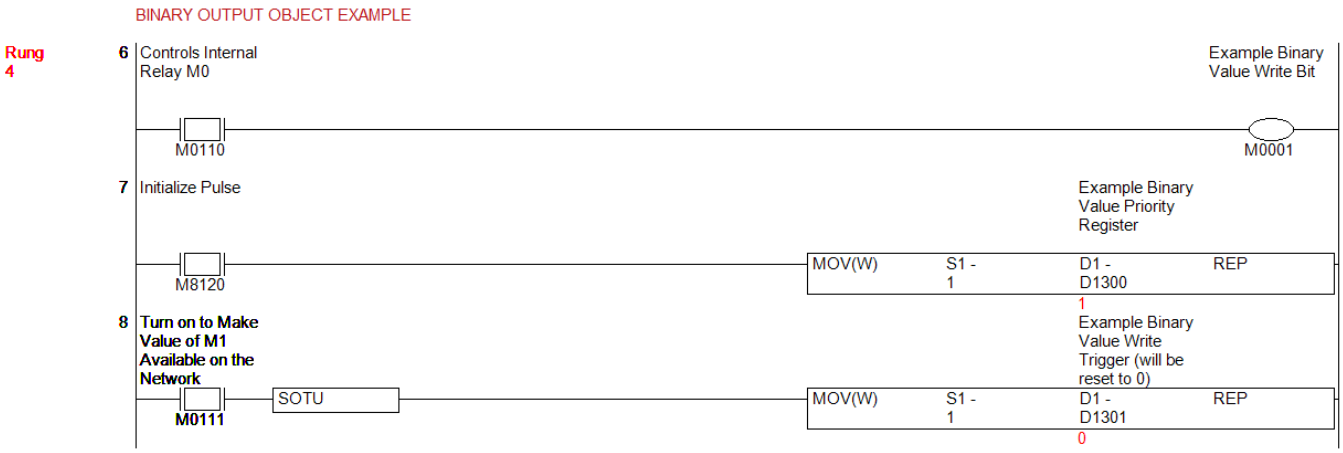


Testing the Binary Value Object

When we created the Binary Value object, we used the following internal relays and data registers:

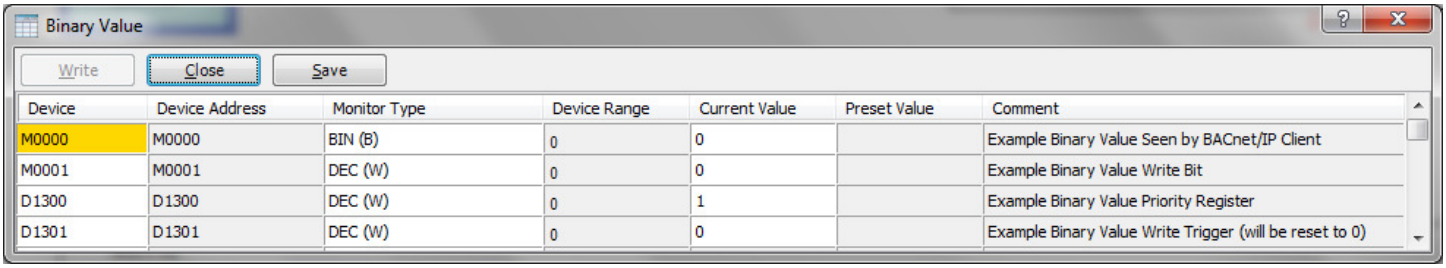


In the PLC ladder program, we have the following logic:



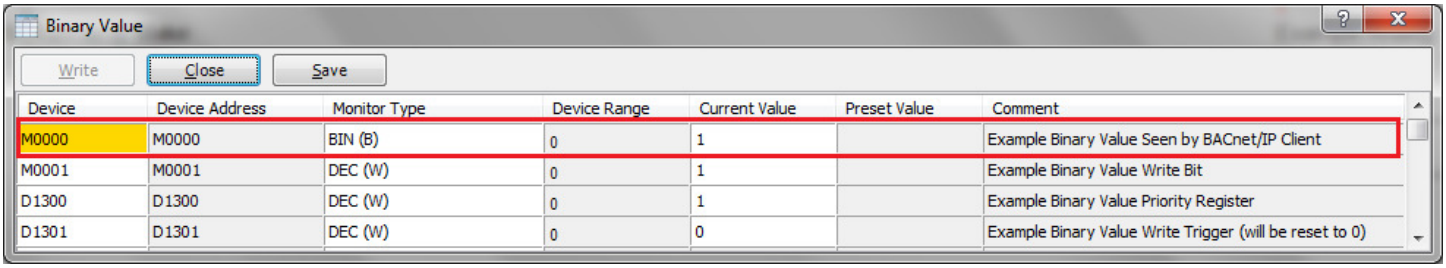
Internal relay M110 controls the status of M1. When we turn M111 on and off, it will make the status of M1 available to the network.

The status that the BACnet/IP client is currently reading is shown in M0.
If you open the Binary Value custom monitor dialog box, you'll see that the BACnet/IP client sees a value of 0:



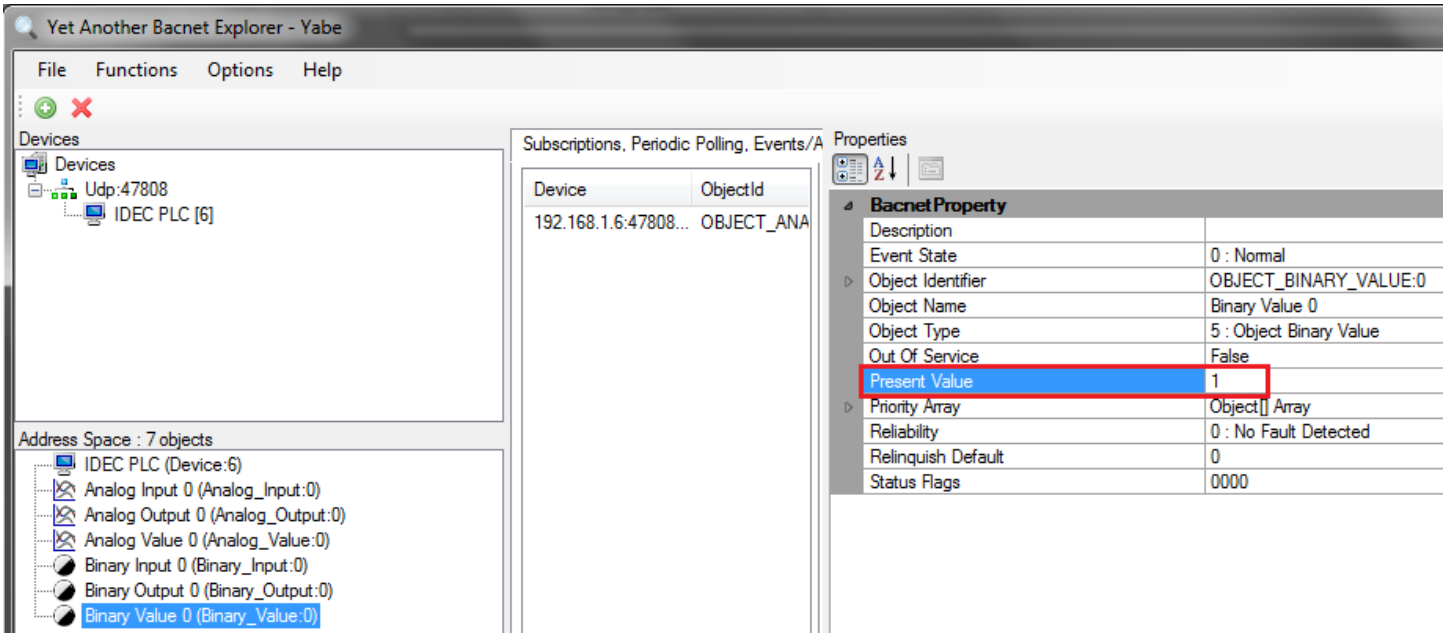
Device	Device Address	Monitor Type	Device Range	Current Value	Preset Value	Comment
M0000	M0000	BIN (B)	0	0		Example Binary Value Seen by BACnet/IP Client
M0001	M0001	DEC (W)	0	0		Example Binary Value Write Bit
D1300	D1300	DEC (W)	0	1		Example Binary Value Priority Register
D1301	D1301	DEC (W)	0	0		Example Binary Value Write Trigger (will be reset to 0)

If we turn M110 on, that will turn M1 on. If we then turn on M111 on and off, that will make the status of M1 available to the network:



Device	Device Address	Monitor Type	Device Range	Current Value	Preset Value	Comment
M0000	M0000	BIN (B)	0	1		Example Binary Value Seen by BACnet/IP Client
M0001	M0001	DEC (W)	0	1		Example Binary Value Write Bit
D1300	D1300	DEC (W)	0	1		Example Binary Value Priority Register
D1301	D1301	DEC (W)	0	0		Example Binary Value Write Trigger (will be reset to 0)

We can also confirm this in YABE by clicking on the Binary Value object and looking at its Present Value:



END