This is a printed copy of the compiled help found in the configuration tool.

BB2-3020, BB2-7020
LonWorks BACnet Gateway
Rev. 1.0 – March 2015

User Guide

Babel Buster 2

User Guide Contents

User Guide

Babel Buster 2

Models BB2-3020, BB2-7020
LonWorks BACnet Gateway
Rev. 1.0 – March 2015

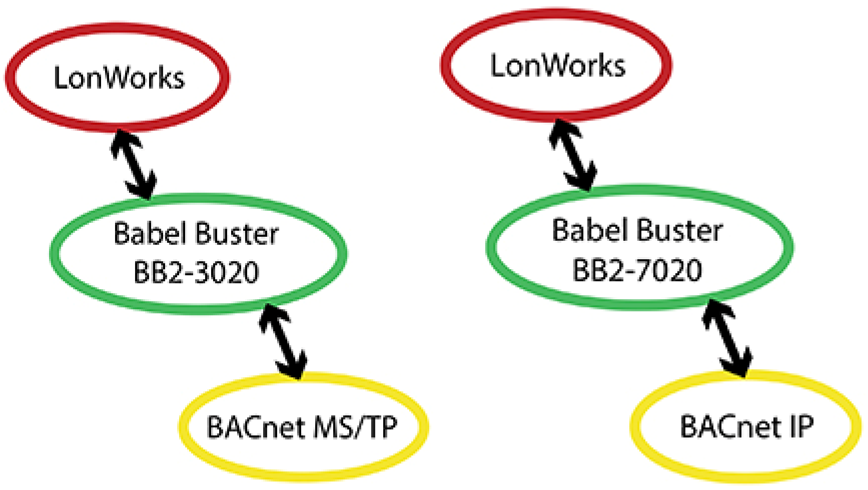
1 Introduction

1.1 How to Use This Guide

The first few sections of this user guide provide background information on how the gateway works, and an overview of the configuration process. The next several sections are guides for each of the tabs found on the screen of the configuration software. The final sections are reference material.

You should at least read the overview sections to gain an understanding of how the gateway functions. You can use the remaining sections as reference material to look up as needed. There is a help icon in the top menu bar of every page in the configuration tool software. Click the help icon (blue button with question mark) at any time to open the section of the user guide that pertains to that page.

1.2 Overview of Gateway Devices



Babel Buster model BB2-3020 is a LonWorks to BACnet MS/TP gateway. It has two processors, an ARM7 and an Echelon FT5000. The FT5000 is running the LonWorks Short Stack microserver, and acts as a LonWorks communications port for the main application running on the ARM7.

Babel Buster model BB2-7020 is a LonWorks to BACnet IP gateway. It adds a third processor (another ARM7, and more importantly, more memory) that provides the Ethernet support for running BACnet IP.

All configuration of all LonWorks gateway models is done via a local USB connection to the gateway. Although the BB2-7020 does have a TCP/IP network connection, it does not have the web server common to certain models of Control Solutions gateways. The complexity of configuration of the LonWorks gateway is not well suited to being web based. By using the USB connection for all versions of the LonWorks gateway, the configuration tool and process is consistent throughout.

1.3 Important Safety Notice

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

1.4 Warranty

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

1.5 Required License Information

The BB2-LON configuration tool includes the SmartWin library (http://smartwinlib.org) under the following terms:

2 Overview of Gateway Functions

2.1 Object Server Model for a Gateway

Control Solutions gateways are not simple protocol translators. It is not possible to do an effective job of simply converting one protocol directly to another. Any attempt to do so would likely have negative effects on the networks on both sides of the gateway. An effective solution requires an intelligent device that can properly and efficiently act as a native device on each network. Control Solutions gateways function as two native devices, one on each network, with a shared data base in between them. They function as clients and/or servers on each network.

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

The concept of data "object" is a universal concept used in all Control Solutions gateways. However, with BACnet, it can get a little confusing because BACnet also calls its data entities "objects". The internal gateway data object and BACnet object are two different things the the Babel Buster gateway. Any given gateway data object can be assigned to any of several BACnet object types. The same gateway data object can be assigned to a BACnet Analog Input or BACnet Binary Output, for example. The "BACnet object" really becomes more of a BACnet address while the gateway data object really holds the data. In this discussion, we also refer to a collection of gateway data objects as simply the data base.

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

Client functionality of a Control Solutions gateway is autonomous. In other words, when acting as a BACnet client, the gateway will continuously poll other BACnet server device(s) on its own, and keep a copy of the most recent data obtained from (or sent to) those BACnet server device(s). LonWorks "clients" may read the data at any time, and write new data to the data base at any time. Most often, the gateway is configured to read server (or slave) devices periodically, and write to the server devices when new data is received from a client.

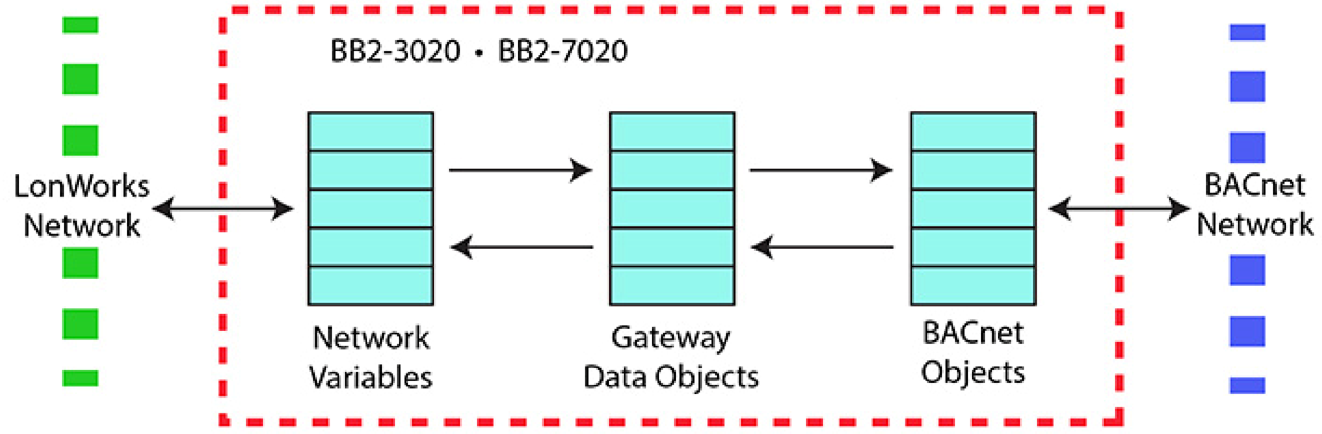
The BB2-3020 and BB2-7020 can each function as both client and server at the same time. BACnet MS/TP and BACnet IP are both effectively peer to peer networks.

2.2 Data Flow in the Gateway

The LonWorks network always interacts with Network Variables (NVs) in any LonWorks device, including the Babel Buster 2 LonWorks Gateways. A network management tool such as Echelon's LonMaker is used to "bind" NVs in the gateway with NVs in other devices.

A Network Variable Input (NVI) means other devices will write data to this NV. The NVI is receiving "input" from the LonWorks network. A Network Variable Output (NVO) means data is being written to ther devices. The NVO is sending "output" to the LonWorks network.

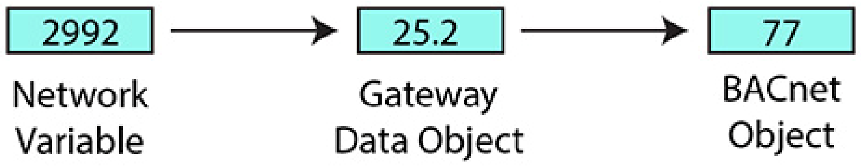
There are three different realizations of the same data within the LonWorks gateway. The direct connection to the LonWorks network is the Network Variable. The direct connection to the BACnet network is the BACnet object. The translating connection in between is a gateway data object.



At first, it may not seem clear why there needs to be these different realizations of the "same" data. The fact that the protocols are incompatible is why we need a gateway in the first place. But why not simply send the data "as is"? The following example is a frequently used translation.

The Network Variable in the diagram below is LonMark type SNVT_temp. The LonMark specification for SNVTs (Standard Network Variable Types) provides the scaling for temperature transmitted over the LonWorks network as SNVT_temp. The raw data is a 16-bit binary number, but scaling provides 0.01 degree resolution. There is also a Kelvin offset in SNVT_temp. As a result, a temperature of 77°F is a raw value of 2992 in the LonWorks NV. That is probably not what your BACnet front end wants to see.

Conversion from LonMark types to standard engineering units is done internally by the LonWorks gateway. When the NV is SNVT_temp, and the internal data object is defined as type 'floating point', the value contained in the internal data object is 25.2°C (when the raw data is 2992). LonWorks values are always metric. The BACnet object mapping provides for further scaling. By applying a scale factor of 1.8 and offset of 32, the Celsius reading now becomes Fahrenheit when provided to BACnet as an Analog Input object.

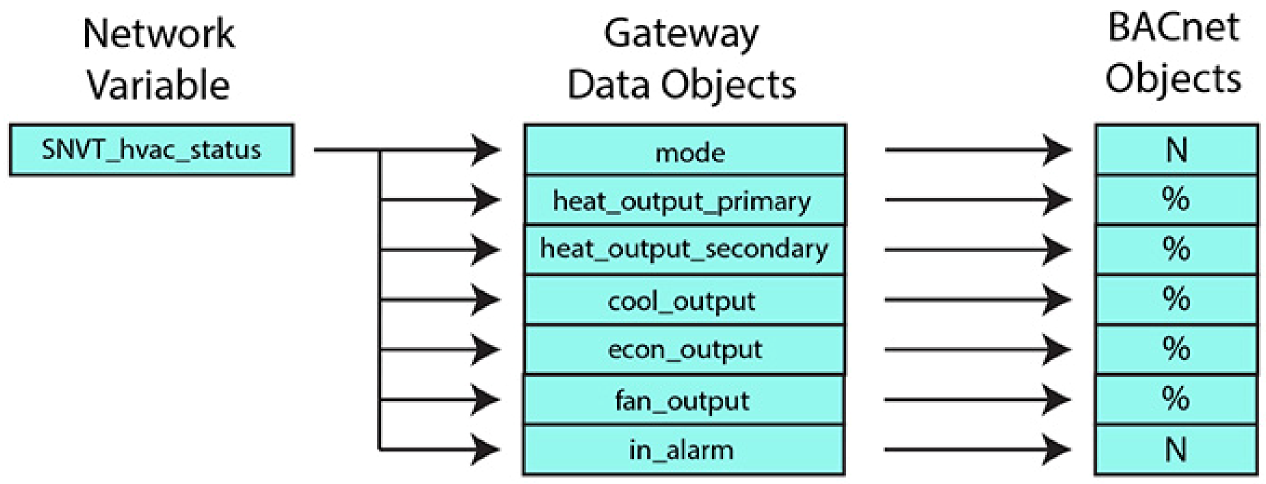


Next we will discuss an even more compelling reason why data cannot simply be sent "as is". Many LonWorks Network Variables are "structure", meaning a single network variable actually contains multiple pieces of data. A BACnet object can contain only a single value. Therefore, it is not possible to do a direct one-to-one translation of these network variables to BACnet objects. In the case of structured data, a single NV translates into multiple BACnet objects, possibly not even of the same type.

A commonly used structured variable is SNVT_hvac_status. It contains seven data fields. These need to be mapped to seven different BACnet objects in order for BACnet to do anything meaningful with the status information. This mapping is partly automated by the configuration software tool provided for the Babel Buster 2 LonWorks Gateways. We say "partly" automated because you do need to select whether or not to include the variable in your mapping, and you also need to enter any applicable scaling on the BACnet side. Scaling for standard NV types will be handled automatically. Scaling for user defined NV types (UNVTs) needs to be entered manually.

The configuration software does not provide any tutorial on what various SNVTs are specified by LonMark. You need to go to the LonMark web site (www.lonmark.org) to obtain a copy of this documentation. If you do not already have a copy, you are strongly encouraged to obtain one since *it is unlikely that you will be able to effectively use any LonWorks gateway without an understanding of the LonWorks*

Network Variables.



2.3 LonWorks Objects and Function Blocks

LonWorks devices are required to operate as a collection of LonWorks objects. All LonWorks devices have, at the very minimum, a Node Object. Beyond that, the type and number of objects is entirely up to the manufacturer. The Babel Buster 2 LonWorks Gateways have a total of 241 objects summarized in the following table.

Object Type	LON Object #	NV index	FB array index	Gateway FB #
Node	0	0 .. 2	n/a	n/a
Open Loop Sensor	1 .. 120	3 .. 122	0 .. 119	OLS 1 .. 120
Open Loop Actuator	121 .. 240	123 .. 242	0 .. 119	OLA 1 .. 120

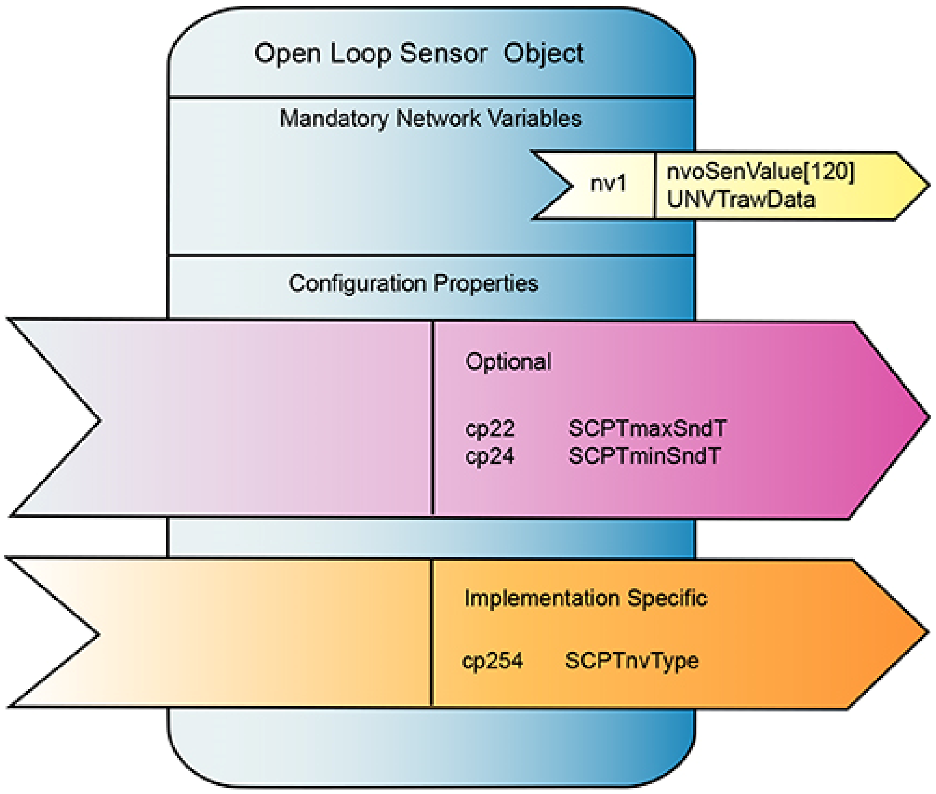
A "LonWorks Object" refers to the function block that encapsulates one or more network variables (NV). In the case of the LonWorks gateways, each function block (FB) contains one network variable. The LonWorks Object number is used when making an object request via the Node Object. Such requests are made by network management tools, especially during network commissioning, and have no use directly related to gateway data translation.

The LonWorks Object and internal gateway "object" are not the same object - they simply use the term "object" to refer to the entity. In the diagrams above, and throughout gateway configuration discussion, the term "Network Variable" is synonymous with LonWorks Object, and "object" refers to the gateway's internal object. Unless the word "object" is preceded by "LonWorks" or "BACnet", the reference to "object" will mean internal gateway data object throughout this user guide.

The NV index is used to look up network variables within the LonWorks NV processing. You will usually have little reason to be concerned about NV index, except that this is the number you will see if you are looking at network traffic on the LonScanner protocol analyzer, or using software such as Nodeutil.exe to directly query NVs.

The FB array index is what you will see in the list of variables when using the LonMaker browser to look at variables and configuration properties in the gateway. The gateway FB number is how you will see the FBs referenced in the configuration software for the gateway.

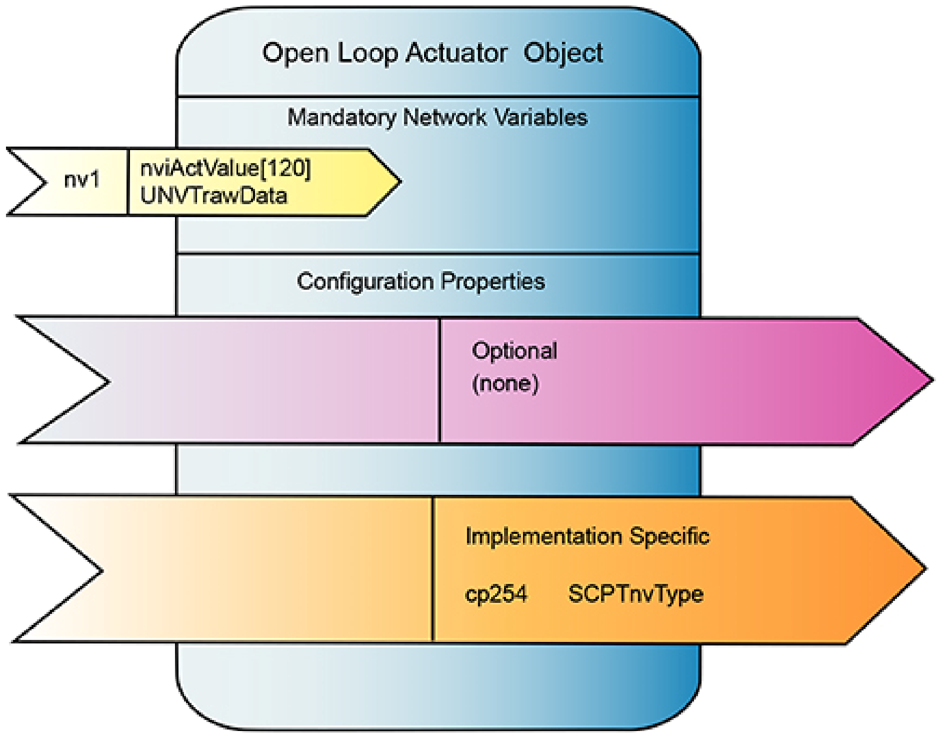
The Open Loop Sensor FB treats BACnet as sensor hardware.



If the gateway a BACnet client, the gateway reads BACnet objects from other server devices.

If the gateway is a BACnet server, it waits for an external client to write to BACnet objects in the gateway.

Either way, when new data is obtained via BACnet, it is propagated to the LonWorks network, specifically to any NVs in other nodes that are bound to this NV.



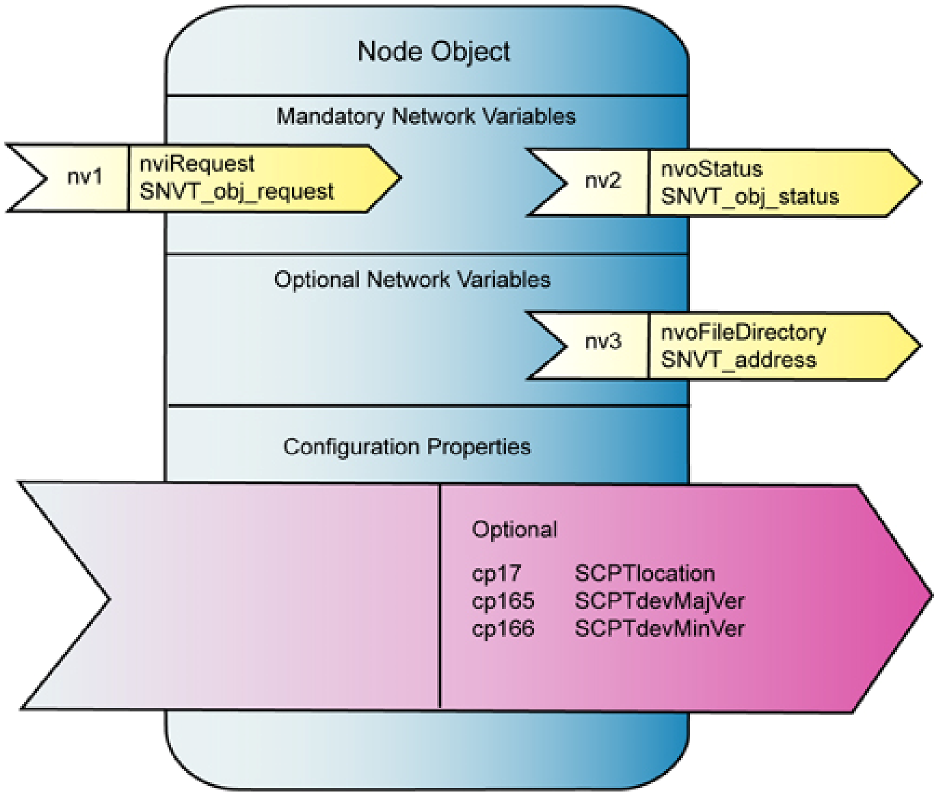
The Open Loop Actuator FB treats BACnet as actuator hardware.

If the gateway is a BACnet client, the gateway writes to BACnet objects in other server devices.

If the gateway is a BACnet server, it waits for the external client to read BACnet objects from the gateway.

Either way, when new data is written to the NV by the LonWorks network (typically by other LonWorks nodes bound to this NV), that data is made available to BACnet.

The Node Object is required primarily as a maintenance feature of the LonWorks device. It is used to enable and disable all other objects in the device, and is used to



check status of other objects. The Node Object also provides the file directory which is necessary in order for the network management tool to access the device's configuration properties.

You normally have no need to be concerned with the Node Object for purposes of configuring the gateway, but this object is critical to network management tools for commissioning the gateway on the LonWorks network.

3 How to Configure the Gateway

3.1 Pick Your Approach

The configuration process, in summary, is a matter of mapping Network Variables to BACnet objects. There are three basic approaches to configuring the gateway:

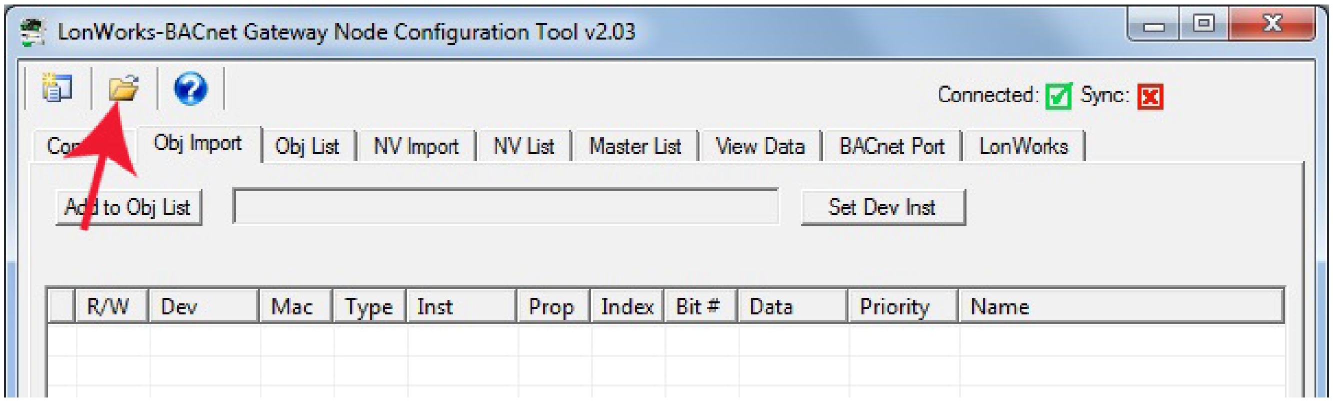
- Start with spreadsheet of BACnet objects and auto-build the LonWorks network variable (NV) set.
- Start with an XIF file for a LonWorks device and auto-build the BACnet object set that best matches it.
- Start from scratch and build configuration entirely manually.

Choose the approach that is most meaningful to you.

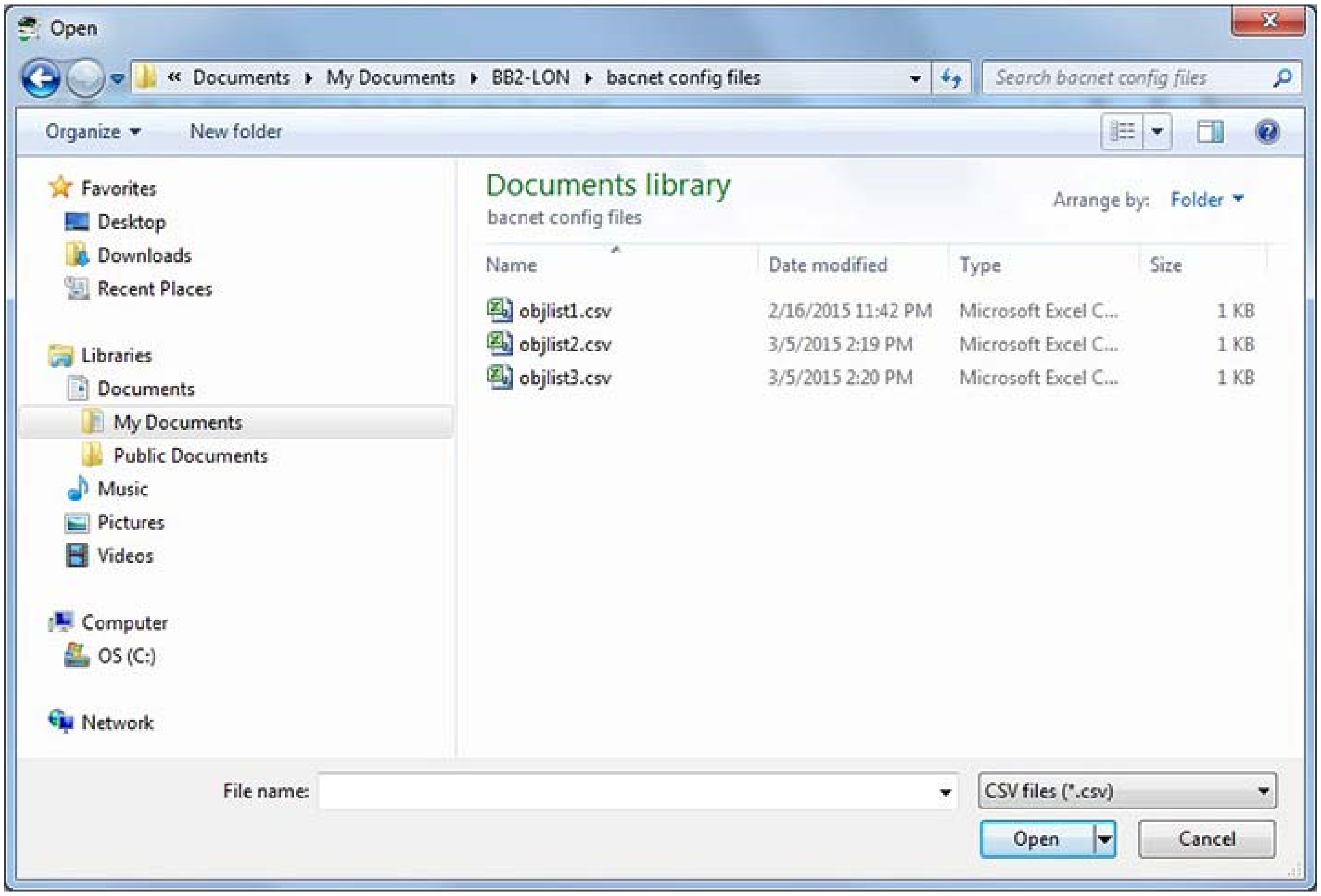
3.2 Build Configuration from CSV List of BACnet Objects

Starting with a BACnet object list for a specific BACnet device, and creating a LonWorks interface for it, is by far the most common application of the LonWorks gateway. You will need to obtain an object list from the manufacturer of the BACnet equipment. If you are lucky, you will receive this in the form of a spreadsheet that you can simply edit. If not, you will need to create a simple spreadsheet and save it in CSV format.

To build your gateway configuration from a CSV file, start by going to the Obj Import page, then click on the file open icon.



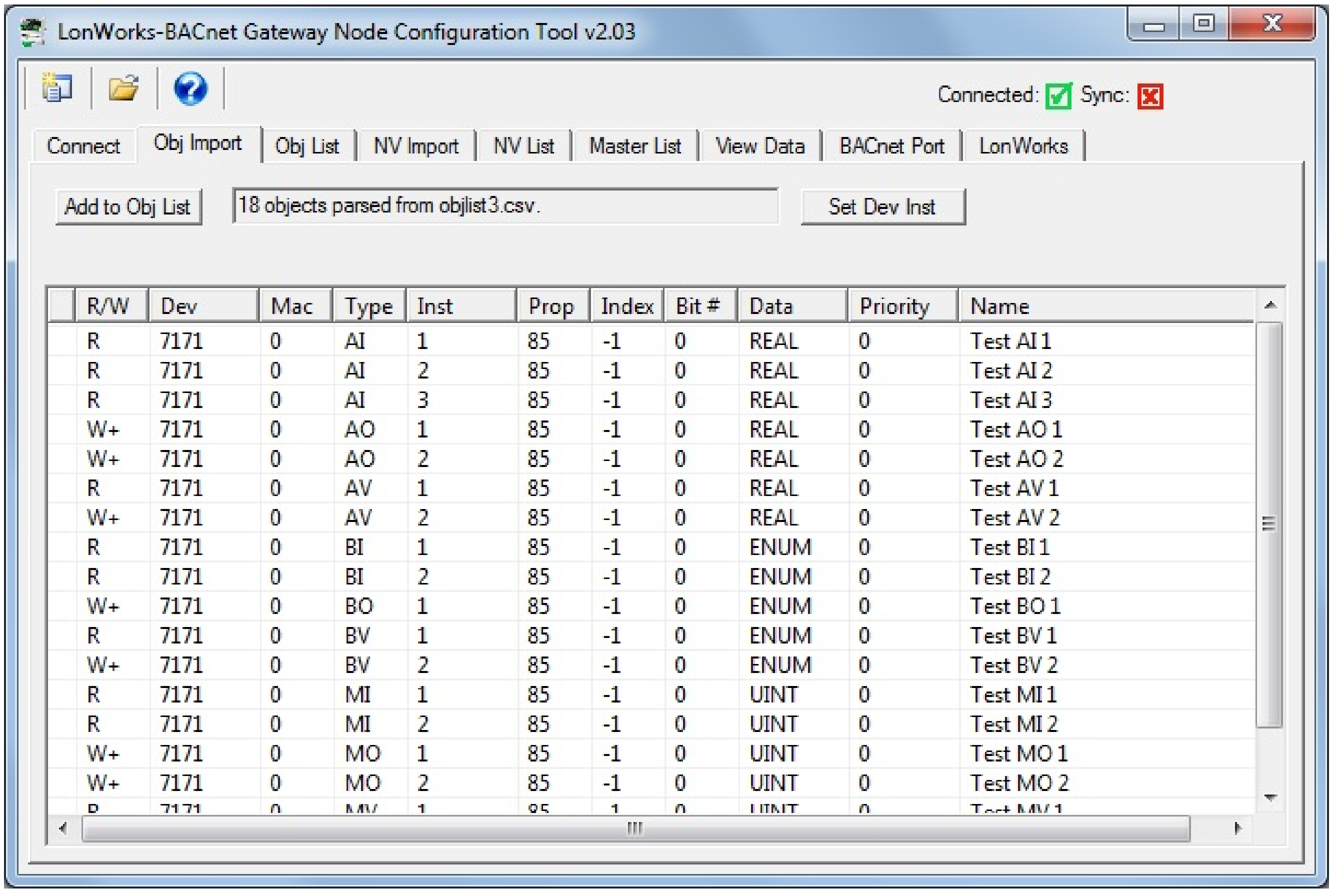
You will see the familiar file open dialog. Select your CSV file.



Name	Date modified	Type	Size
objlist1.csv	2/16/2015 11:42 PM	Microsoft Excel C...	1 KB
objlist2.csv	3/5/2015 2:19 PM	Microsoft Excel C...	1 KB
objlist3.csv	3/5/2015 2:20 PM	Microsoft Excel C...	1 KB

For reference, the content of the objlist3.csv file loaded here contains the following CSV list, and upon loading, will appear as shown in the next screen shot.

```
RW,DEV,OBJTYPE,OBJINST,PROP,DATA,NAME
R,7171,AI,1,85,REAL,Test AI 1
R,7171,AI,2,85,REAL,Test AI 2
R,7171,AI,3,85,REAL,Test AI 3
W+,7171,AO,1,85,REAL,Test AO 1
W+,7171,AO,2,85,REAL,Test AO 2
R,7171,AV,1,85,REAL,Test AV 1
W+,7171,AV,2,85,REAL,Test AV 2
R,7171,BI,1,85,ENUM,Test BI 1
R,7171,BI,2,85,ENUM,Test BI 2
W+,7171,BO,1,85,ENUM,Test BO 1
R,7171,BV,1,85,ENUM,Test BV 1
W+,7171,BV,2,85,ENUM,Test BV 2
R,7171,MI,1,85,UINT,Test MI 1
R,7171,MI,2,85,UINT,Test MI 2
W+,7171,MO,1,85,UINT,Test MO 1
W+,7171,MO,2,85,UINT,Test MO 2
R,7171,MV,1,85,UINT,Test MV 1
W+,7171,MV,2,85,UINT,Test MV 2
```



18 objects parsed from objlist3.csv.

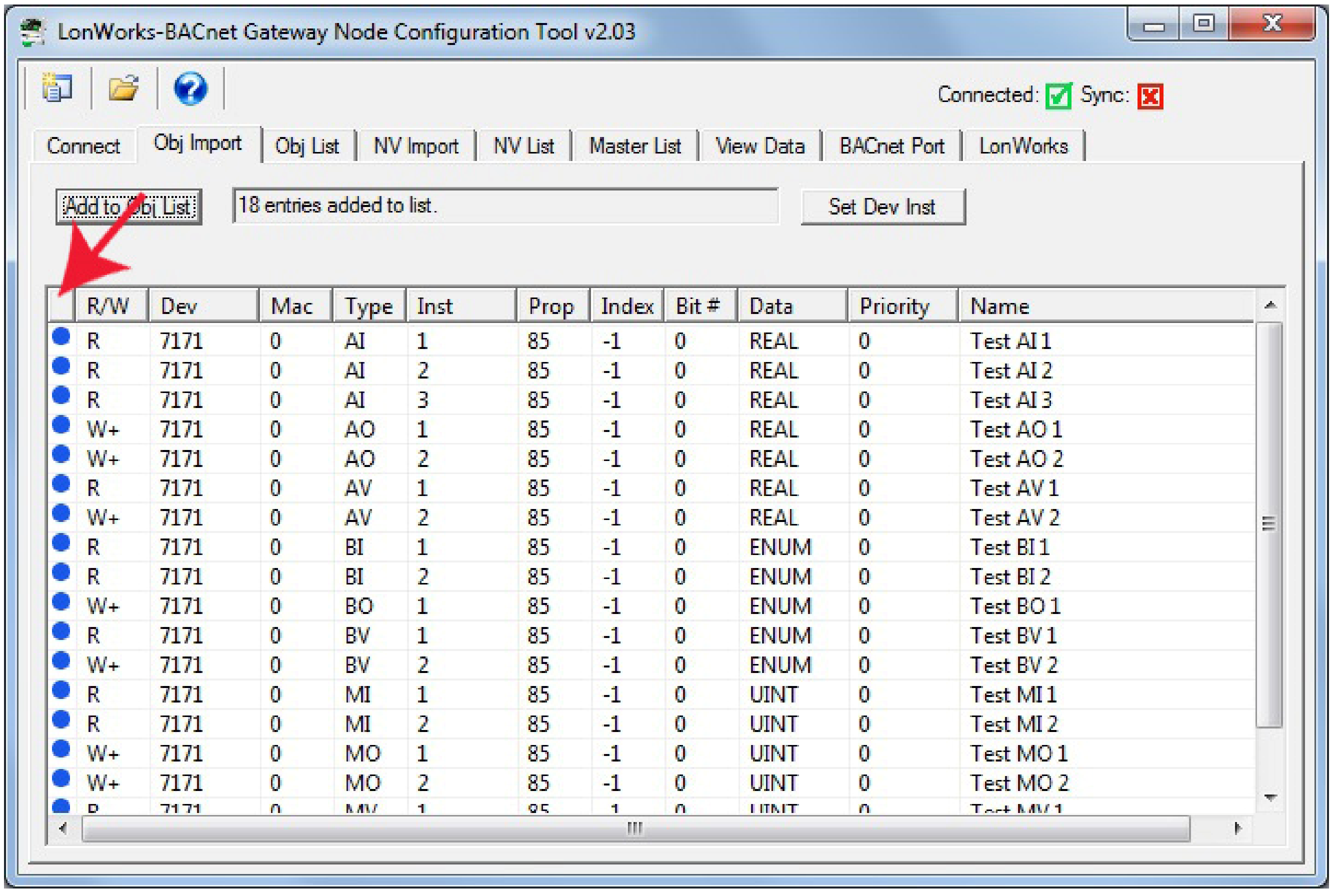
R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
R	7171	0	AI	1	85	-1	0	REAL	0	Test AI 1
R	7171	0	AI	2	85	-1	0	REAL	0	Test AI 2
R	7171	0	AI	3	85	-1	0	REAL	0	Test AI 3
W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO 1
W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO 2
R	7171	0	AV	1	85	-1	0	REAL	0	Test AV 1
W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV 2
R	7171	0	BI	1	85	-1	0	ENUM	0	Test BI 1
R	7171	0	BI	2	85	-1	0	ENUM	0	Test BI 2
W+	7171	0	BO	1	85	-1	0	ENUM	0	Test BO 1
R	7171	0	BV	1	85	-1	0	ENUM	0	Test BV 1
W+	7171	0	BV	2	85	-1	0	ENUM	0	Test BV 2
R	7171	0	MI	1	85	-1	0	UINT	0	Test MI 1
R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1

At this point, the object list is contained in what amounts to a "scratch pad". The CSV file is imported into an interm list so that you can choose which objects you want to include in your configuration. If you created the CSV file, you most likely want all of them. But if somebody else sent you a CSV file containing every object in their BACnet device, you will usually want to select only those that are pertinent to your application.

To select all objects, click the icon column header as illustrated by the red arrow below. This will cause the blue dot to appear on each line, which indicates that this object has been selected to be included in the configuration. To select only certain objects, click the icon area of only those rows you wish to include.

Once you have selected the objects, click the "Add to Obj List" button. Until you click this button, you have no objects in the configured object list. You may, at this point, import another CSV and continue to add multiple objects from multiple CSV files.

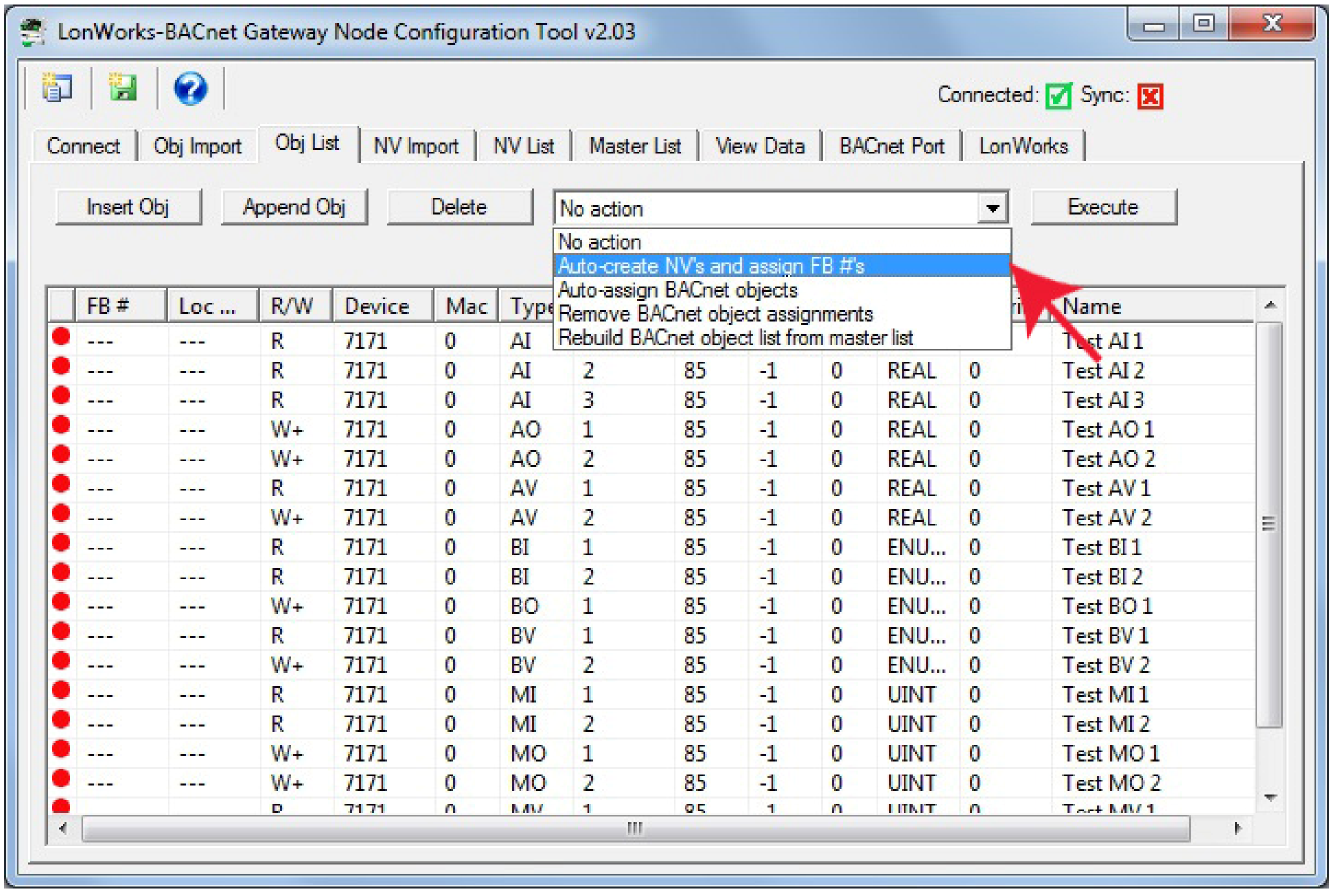
If you will be connecting two or more of the same type of BACnet device, each having identical object sets, click "Add to Obj List". Then use "Set Dev Inst" to select the next device's instance (identifier that will be sent out in a Who-Is message). Now click "Add to Obj List" again. This adds the same set of objects a second time, but with a different device instance the second time.



18 entries added to list.

	R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
●	R	7171	0	AI	1	85	-1	0	REAL	0	Test AI 1
●	R	7171	0	AI	2	85	-1	0	REAL	0	Test AI 2
●	R	7171	0	AI	3	85	-1	0	REAL	0	Test AI 3
●	W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO 1
●	W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO 2
●	R	7171	0	AV	1	85	-1	0	REAL	0	Test AV 1
●	W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV 2
●	R	7171	0	BI	1	85	-1	0	ENUM	0	Test BI 1
●	R	7171	0	BI	2	85	-1	0	ENUM	0	Test BI 2
●	W+	7171	0	BO	1	85	-1	0	ENUM	0	Test BO 1
●	R	7171	0	BV	1	85	-1	0	ENUM	0	Test BV 1
●	W+	7171	0	BV	2	85	-1	0	ENUM	0	Test BV 2
●	R	7171	0	MI	1	85	-1	0	UINT	0	Test MI 1
●	R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
●	W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
●	W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
●	R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1

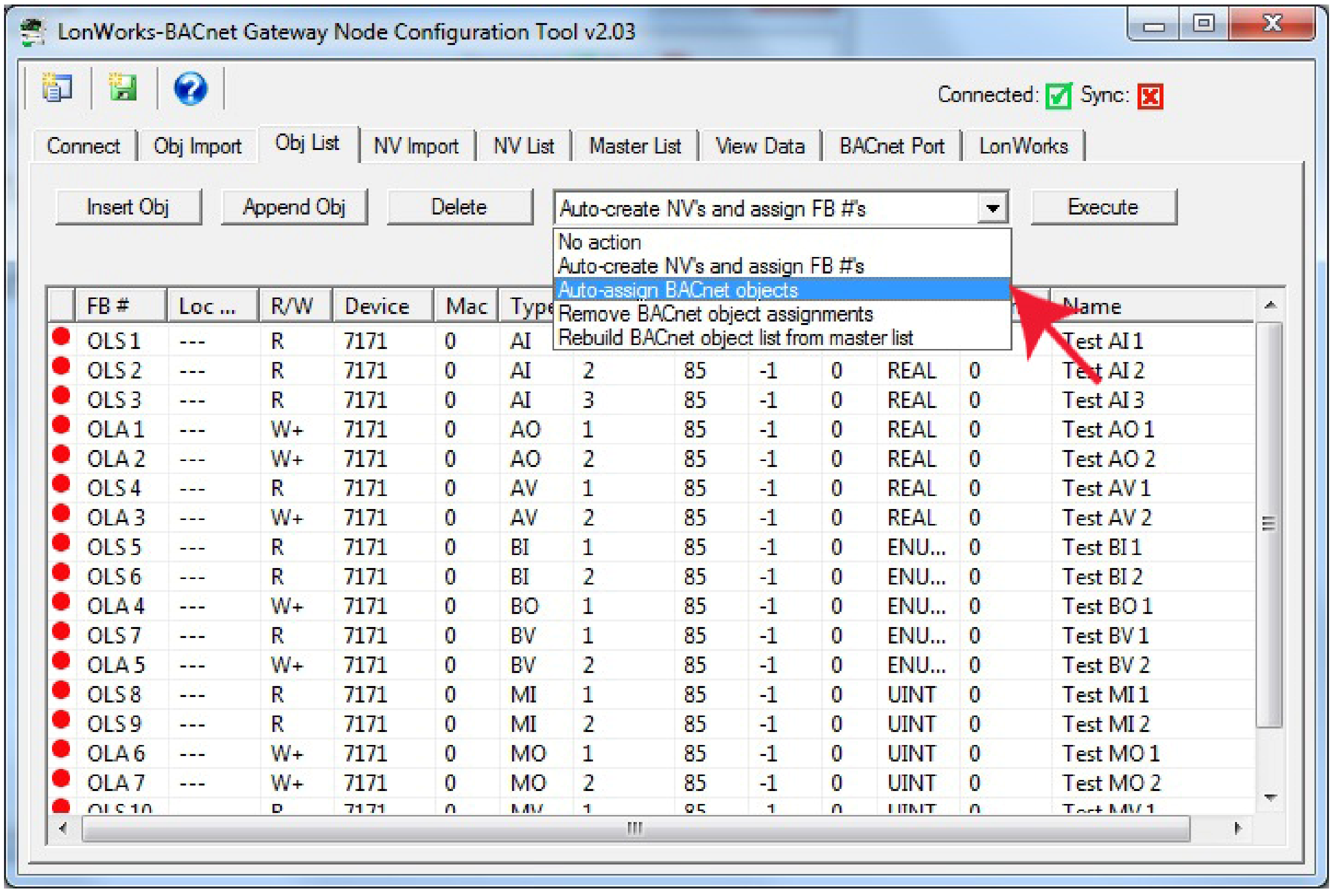
Once you have imported your object list(s), go to the Obj List page. Here is where you begin the auto-build of the rest of your configuration. Start by selecting "Auto-create NV's and assign FB #'s". Then click the Execute button.


LonWorks-BACnet Gateway Node Configuration Tool v2.03
Connected: Sync:
Connect
Obj Import
Obj List
NV Import
NV List
Master List
View Data
BACnet Port
LonWorks
Insert Obj
Append Obj
Delete
No action
Execute
No action
Auto-create NV's and assign FB #'s
Auto-assign BACnet objects
Remove BACnet object assignments
Rebuild BACnet object list from master list

	FB #	Loc ...	R/W	Device	Mac	Type						ri	Name
●	---	---	R	7171	0	AI							Test AI 1
●	---	---	R	7171	0	AI	2	85	-1	0	REAL	0	Test AI 2
●	---	---	R	7171	0	AI	3	85	-1	0	REAL	0	Test AI 3
●	---	---	W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO 1
●	---	---	W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO 2
●	---	---	R	7171	0	AV	1	85	-1	0	REAL	0	Test AV 1
●	---	---	W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV 2
●	---	---	R	7171	0	BI	1	85	-1	0	ENU...	0	Test BI 1
●	---	---	R	7171	0	BI	2	85	-1	0	ENU...	0	Test BI 2
●	---	---	W+	7171	0	BO	1	85	-1	0	ENU...	0	Test BO 1
●	---	---	R	7171	0	BV	1	85	-1	0	ENU...	0	Test BV 1
●	---	---	W+	7171	0	BV	2	85	-1	0	ENU...	0	Test BV 2
●	---	---	R	7171	0	MI	1	85	-1	0	UINT	0	Test MI 1
●	---	---	R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
●	---	---	W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
●	---	---	W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
●			R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1

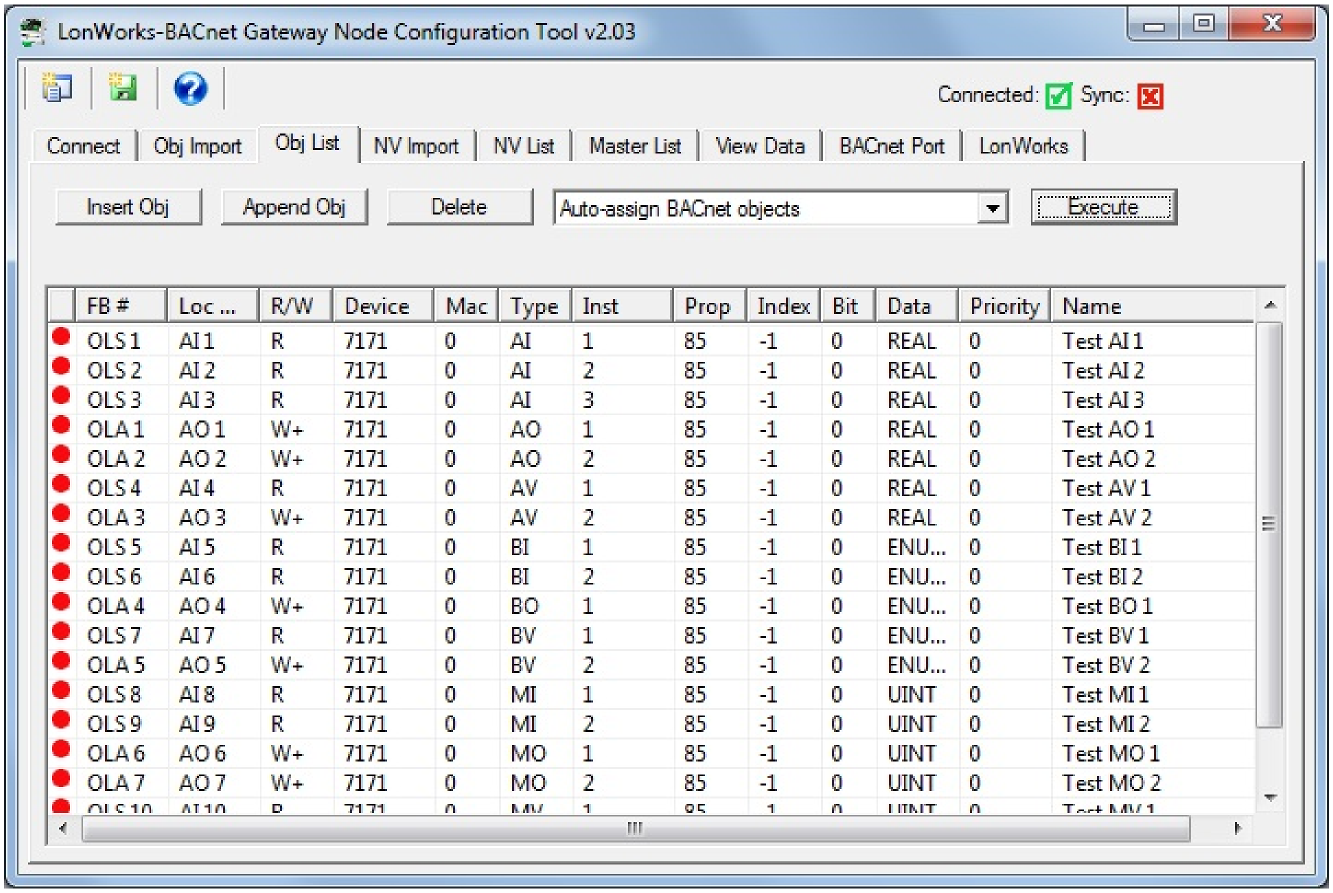
Upon executing the Auto-create NV's, the FB# column will be populated. BACnet objects designated as "W" for write, meaning the gateway will write to these objects in another BACnet device (assuming gateway is client), will be assigned to an Open Loop Actuator function block having a Network Variable Input (NVI). BACnet objects designated as "R" for read, meaning the gateway will read these objects from another BACnet device, will be assigned to an Open Loop Sensor function block having a Network Variable Output (NVO).

Next, select "Auto-assign data objects" from the list and click Execute.


LonWorks-BACnet Gateway Node Configuration Tool v2.03
Connected: ✓ Sync: ✗
Connect
Obj Import
Obj List
NV Import
NV List
Master List
View Data
BACnet Port
LonWorks
Insert Obj
Append Obj
Delete
Auto-create NV's and assign FB #'s
Execute
No action
Auto-create NV's and assign FB #'s
Auto-assign BACnet objects
Remove BACnet object assignments
Rebuild BACnet object list from master list

	FB #	Loc ...	R/W	Device	Mac	Type								Name
●	OLS 1	---	R	7171	0	AI								Test AI 1
●	OLS 2	---	R	7171	0	AI	2	85	-1	0	REAL	0		Test AI 2
●	OLS 3	---	R	7171	0	AI	3	85	-1	0	REAL	0		Test AI 3
●	OLA 1	---	W+	7171	0	AO	1	85	-1	0	REAL	0		Test AO 1
●	OLA 2	---	W+	7171	0	AO	2	85	-1	0	REAL	0		Test AO 2
●	OLS 4	---	R	7171	0	AV	1	85	-1	0	REAL	0		Test AV 1
●	OLA 3	---	W+	7171	0	AV	2	85	-1	0	REAL	0		Test AV 2
●	OLS 5	---	R	7171	0	BI	1	85	-1	0	ENU...	0		Test BI 1
●	OLS 6	---	R	7171	0	BI	2	85	-1	0	ENU...	0		Test BI 2
●	OLA 4	---	W+	7171	0	BO	1	85	-1	0	ENU...	0		Test BO 1
●	OLS 7	---	R	7171	0	BV	1	85	-1	0	ENU...	0		Test BV 1
●	OLA 5	---	W+	7171	0	BV	2	85	-1	0	ENU...	0		Test BV 2
●	OLS 8	---	R	7171	0	MI	1	85	-1	0	UINT	0		Test MI 1
●	OLS 9	---	R	7171	0	MI	2	85	-1	0	UINT	0		Test MI 2
●	OLA 6	---	W+	7171	0	MO	1	85	-1	0	UINT	0		Test MO 1
●	OLA 7	---	W+	7171	0	MO	2	85	-1	0	UINT	0		Test MO 2
●	OLS 10		R	7171	0	MV	1	85	-1	0	UINT	0		Test MV 1

Upon executing Auto-assign data objects, the object numbers allocated will appear in the Obj column.

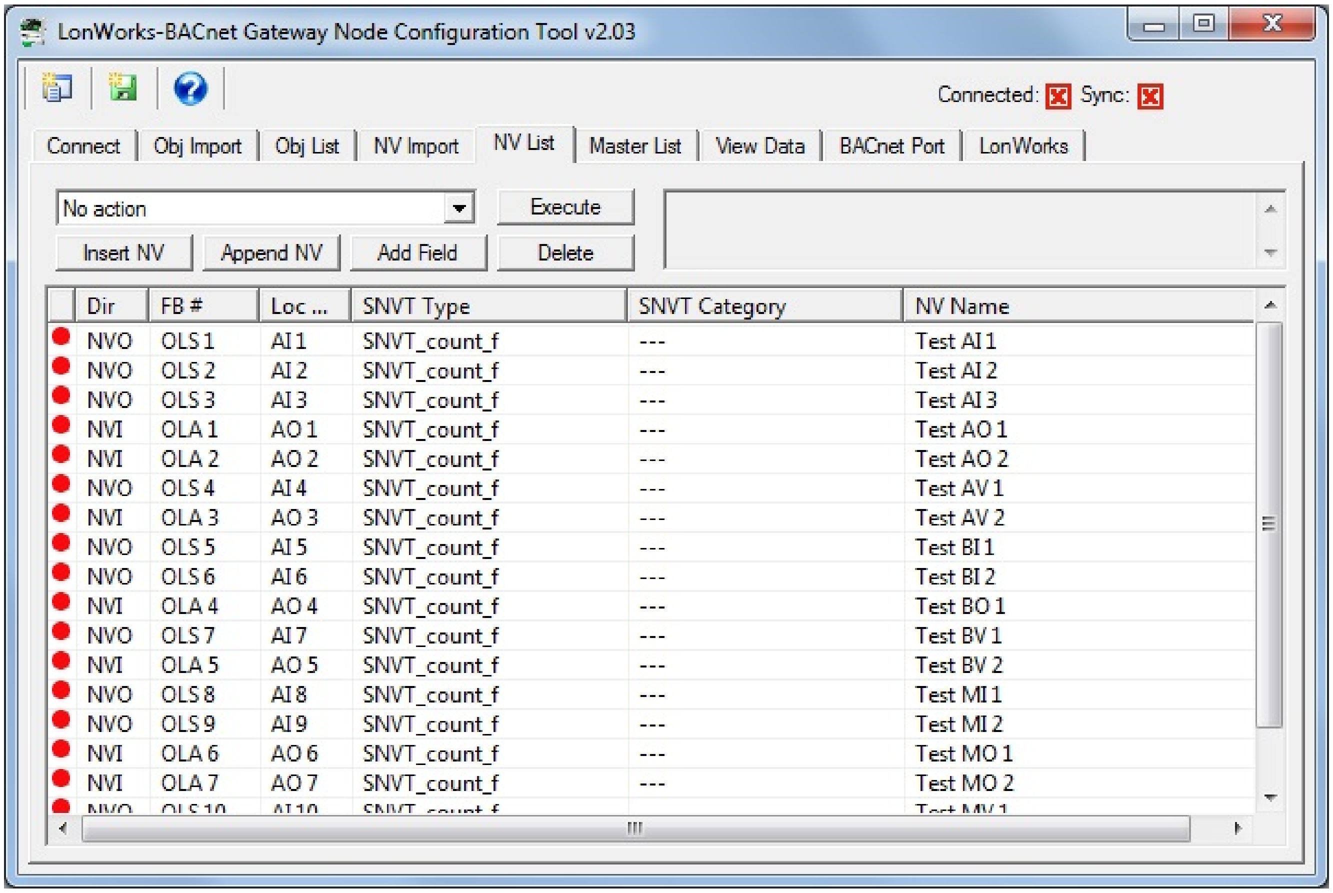


	FB #	Loc ...	R/W	Device	Mac	Type	Inst	Prop	Index	Bit	Data	Priority	Name
●	OLS 1	AI 1	R	7171	0	AI	1	85	-1	0	REAL	0	Test AI 1
●	OLS 2	AI 2	R	7171	0	AI	2	85	-1	0	REAL	0	Test AI 2
●	OLS 3	AI 3	R	7171	0	AI	3	85	-1	0	REAL	0	Test AI 3
●	OLA 1	AO 1	W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO 1
●	OLA 2	AO 2	W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO 2
●	OLS 4	AI 4	R	7171	0	AV	1	85	-1	0	REAL	0	Test AV 1
●	OLA 3	AO 3	W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV 2
●	OLS 5	AI 5	R	7171	0	BI	1	85	-1	0	ENU...	0	Test BI 1
●	OLS 6	AI 6	R	7171	0	BI	2	85	-1	0	ENU...	0	Test BI 2
●	OLA 4	AO 4	W+	7171	0	BO	1	85	-1	0	ENU...	0	Test BO 1
●	OLS 7	AI 7	R	7171	0	BV	1	85	-1	0	ENU...	0	Test BV 1
●	OLA 5	AO 5	W+	7171	0	BV	2	85	-1	0	ENU...	0	Test BV 2
●	OLS 8	AI 8	R	7171	0	MI	1	85	-1	0	UINT	0	Test MI 1
●	OLS 9	AI 9	R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
●	OLA 6	AO 6	W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
●	OLA 7	AO 7	W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
●	OLS 10	AI 10	R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1

Creating of the configuration is now largely complete. You can make alterations to things like data scaling at this point if you wish. You can also change network variable types at this point. But for simply putting your BACnet device on the LonWorks network using generic counts as the network variable type, the configuration is complete. All that remains now is to send the configuration to the gateway device. At this point, the configuration only exists on your PC. It needs to get written into the device before the gateway will be functional.

Go to the NV List page. This is the set of network variables that were automatically created for you. If you wish to export an XIF file defining your LonWorks device, you may do so here by clicking the green file icon to create a new file. Some network management tools such as LonMaker can import the XIF directly from the device and you do not need an XIF file. However, some network management tools do require the XIF file, and this is where you create it.

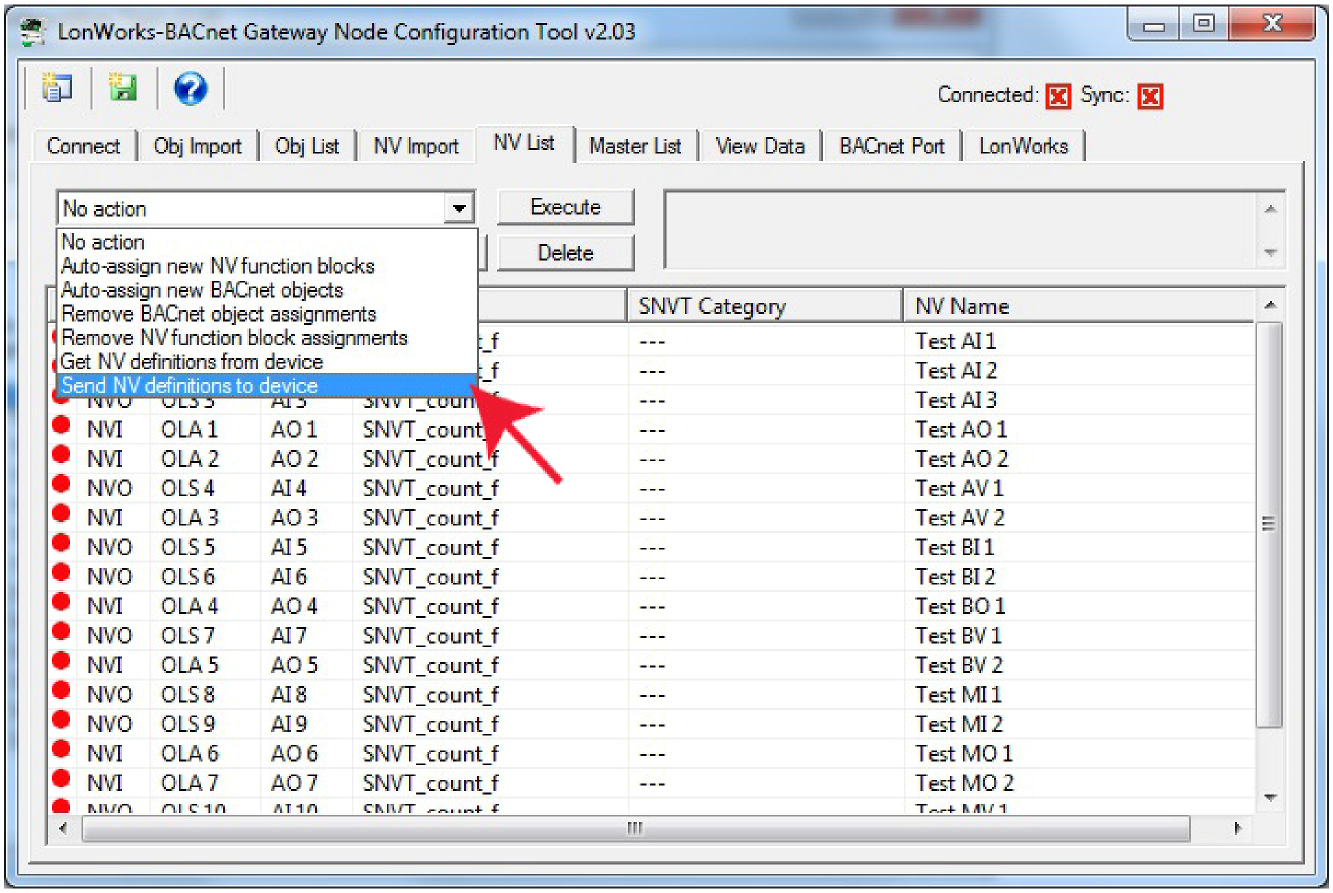
Note: If you want your LonWorks interface to have network variable types other than those that were selected by default, STOP HERE and make those changes. Refer to the NV List page section in this user guide for more detail on changing NV types.



	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVO	OLS 1	AI 1	SNVT_count_f	---	Test AI 1
●	NVO	OLS 2	AI 2	SNVT_count_f	---	Test AI 2
●	NVO	OLS 3	AI 3	SNVT_count_f	---	Test AI 3
●	NVI	OLA 1	AO 1	SNVT_count_f	---	Test AO 1
●	NVI	OLA 2	AO 2	SNVT_count_f	---	Test AO 2
●	NVO	OLS 4	AI 4	SNVT_count_f	---	Test AV 1
●	NVI	OLA 3	AO 3	SNVT_count_f	---	Test AV 2
●	NVO	OLS 5	AI 5	SNVT_count_f	---	Test BI 1
●	NVO	OLS 6	AI 6	SNVT_count_f	---	Test BI 2
●	NVI	OLA 4	AO 4	SNVT_count_f	---	Test BO 1
●	NVO	OLS 7	AI 7	SNVT_count_f	---	Test BV 1
●	NVI	OLA 5	AO 5	SNVT_count_f	---	Test BV 2
●	NVO	OLS 8	AI 8	SNVT_count_f	---	Test MI 1
●	NVO	OLS 9	AI 9	SNVT_count_f	---	Test MI 2
●	NVI	OLA 6	AO 6	SNVT_count_f	---	Test MO 1
●	NVI	OLA 7	AO 7	SNVT_count_f	---	Test MO 2
●	NVO	OLS 10	AI 10	SNVT_count_f		Test MV 1

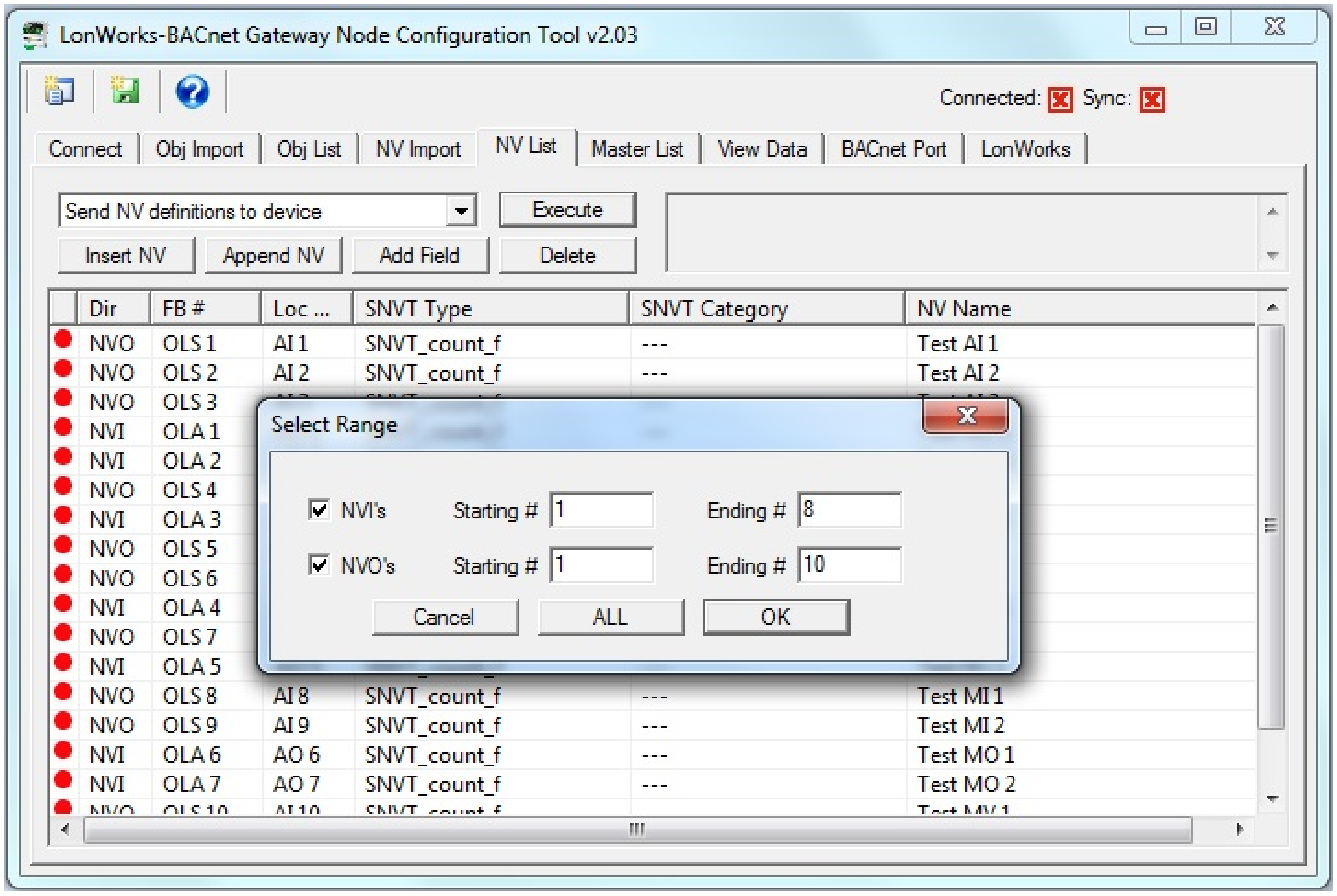
There are two steps in writing the configuration to the gateway device. The first step is to send the network variable (NV) list to the gateway. The second step is to send the data objects (which include the BACnet object mapping) to the gateway.

To send the NV list to the gateway, select "Send NV definitions to device" from the list and click Execute.


LonWorks-BACnet Gateway Node Configuration Tool v2.03
Connected:
Sync:
Connect
Obj Import
Obj List
NV Import
NV List
Master List
View Data
BACnet Port
LonWorks
No action
Execute
Delete
No action
Auto-assign new NV function blocks
Auto-assign new BACnet objects
Remove BACnet object assignments
Remove NV function block assignments
Get NV definitions from device
Send NV definitions to device

					SNVT Category	NV Name
				_f	---	Test AI 1
				_f	---	Test AI 2
●	NVO	OLS 3	AI 3	SNVT_coun	---	Test AI 3
●	NVI	OLA 1	AO 1	SNVT_count	---	Test AO 1
●	NVI	OLA 2	AO 2	SNVT_count_f	---	Test AO 2
●	NVO	OLS 4	AI 4	SNVT_count_f	---	Test AV 1
●	NVI	OLA 3	AO 3	SNVT_count_f	---	Test AV 2
●	NVO	OLS 5	AI 5	SNVT_count_f	---	Test BI 1
●	NVO	OLS 6	AI 6	SNVT_count_f	---	Test BI 2
●	NVI	OLA 4	AO 4	SNVT_count_f	---	Test BO 1
●	NVO	OLS 7	AI 7	SNVT_count_f	---	Test BV 1
●	NVI	OLA 5	AO 5	SNVT_count_f	---	Test BV 2
●	NVO	OLS 8	AI 8	SNVT_count_f	---	Test MI 1
●	NVO	OLS 9	AI 9	SNVT_count_f	---	Test MI 2
●	NVI	OLA 6	AO 6	SNVT_count_f	---	Test MO 1
●	NVI	OLA 7	AO 7	SNVT_count_f	---	Test MO 2
●	NVO	OLS 10	AI 10	SNVT_count_f		Test MV 1

When you click Execute to send the NV list, it will scan the configuration and see how many NV entries are actually used. This will appear in the Select Range dialog. Generally, with a newly created configuration, you would simply click OK. If you later make changes and need to re-send those changes, you can alter the range to send only those items changed.

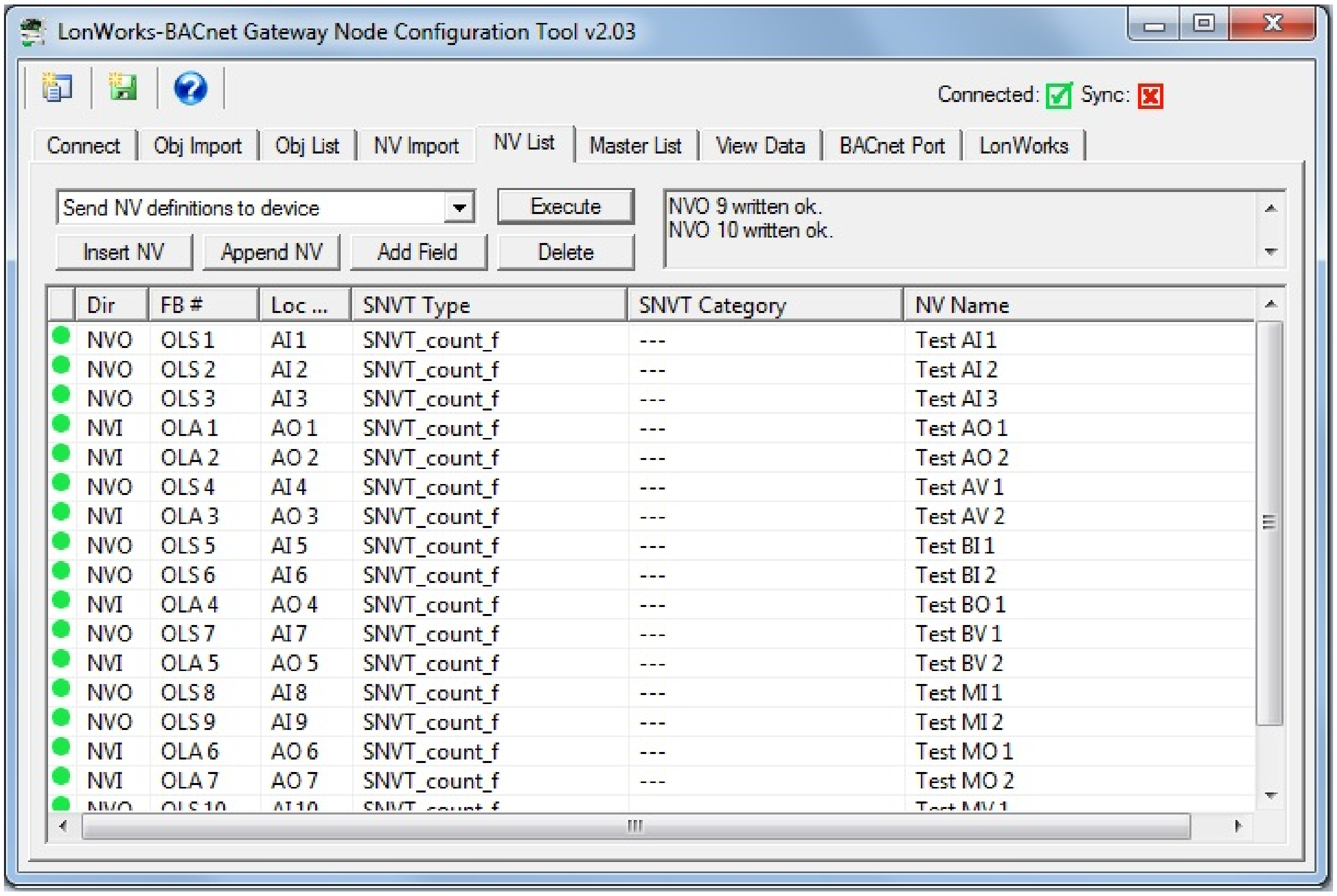


	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVO	OLS 1	AI 1	SNVT_count_f	---	Test AI 1
●	NVO	OLS 2	AI 2	SNVT_count_f	---	Test AI 2
●	NVO	OLS 3				
●	NVI	OLA 1				
●	NVI	OLA 2				
●	NVO	OLS 4				
●	NVI	OLA 3				
●	NVO	OLS 5				
●	NVO	OLS 6				
●	NVI	OLA 4				
●	NVO	OLS 7				
●	NVI	OLA 5				
●	NVO	OLS 8	AI 8	SNVT_count_f	---	Test MI 1
●	NVO	OLS 9	AI 9	SNVT_count_f	---	Test MI 2
●	NVI	OLA 6	AO 6	SNVT_count_f	---	Test MO 1
●	NVI	OLA 7	AO 7	SNVT_count_f	---	Test MO 2
●	NVO	OLS 10	AI 10	SNVT_count_f		Test MV 1

Select Range

✓ NVI's	Starting #	1	Ending #	8
✓ NVO's	Starting #	1	Ending #	10

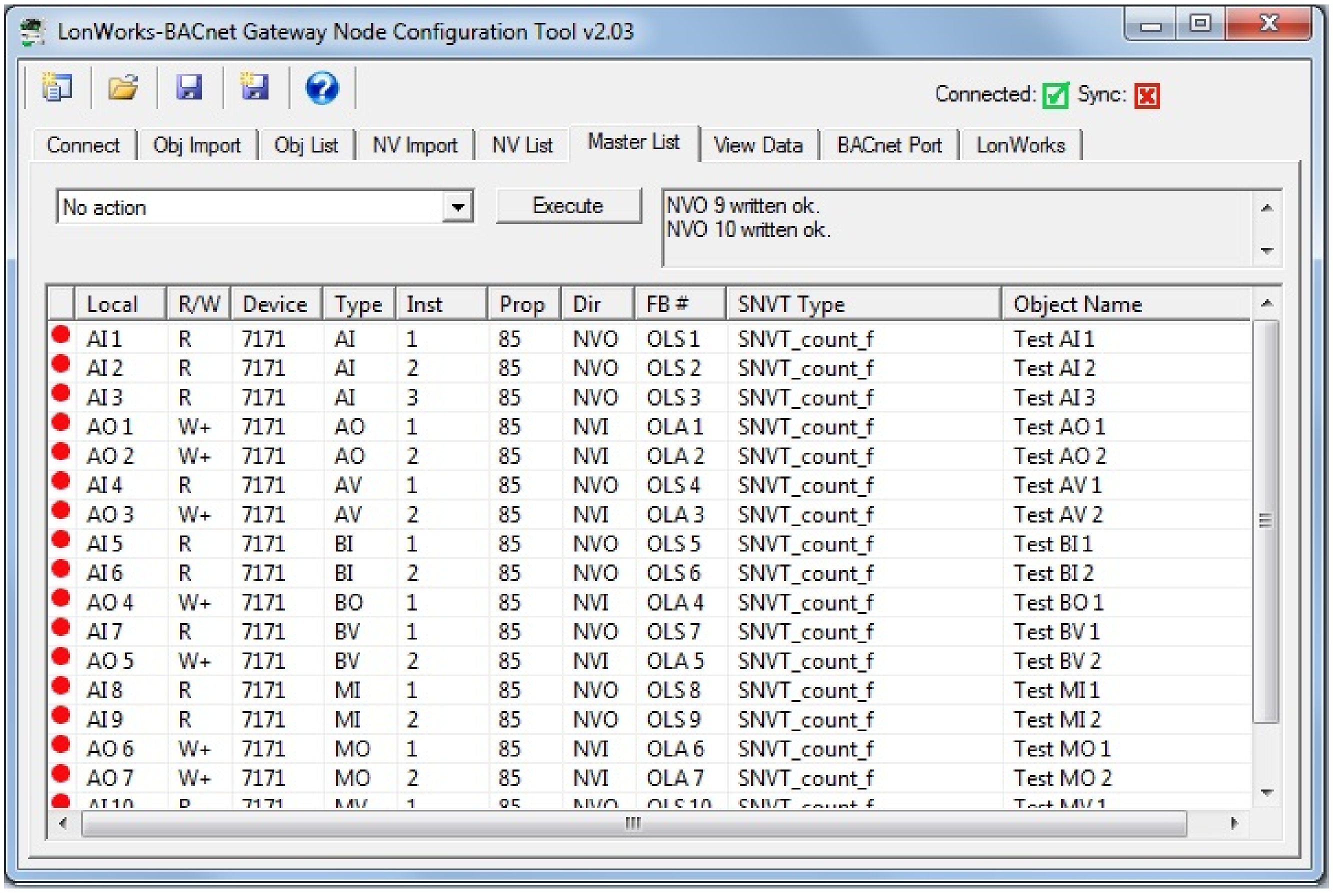
The red dot in the icon column means this line of configuration information has not yet been sent to the gateway device. After executing the Send NV definitions to device, the icons will turn green indicating that the tool software and the device are now in sync and information displayed on the screen is also contained in the device. If you make any changes, the icon will turn red again for the line that was changed. You would then need to re-send.



```
NVO 9 written ok.
NVO 10 written ok.
```

	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVO	OLS 1	AI 1	SNVT_count_f	---	Test AI 1
●	NVO	OLS 2	AI 2	SNVT_count_f	---	Test AI 2
●	NVO	OLS 3	AI 3	SNVT_count_f	---	Test AI 3
●	NVI	OLA 1	AO 1	SNVT_count_f	---	Test AO 1
●	NVI	OLA 2	AO 2	SNVT_count_f	---	Test AO 2
●	NVO	OLS 4	AI 4	SNVT_count_f	---	Test AV 1
●	NVI	OLA 3	AO 3	SNVT_count_f	---	Test AV 2
●	NVO	OLS 5	AI 5	SNVT_count_f	---	Test BI 1
●	NVO	OLS 6	AI 6	SNVT_count_f	---	Test BI 2
●	NVI	OLA 4	AO 4	SNVT_count_f	---	Test BO 1
●	NVO	OLS 7	AI 7	SNVT_count_f	---	Test BV 1
●	NVI	OLA 5	AO 5	SNVT_count_f	---	Test BV 2
●	NVO	OLS 8	AI 8	SNVT_count_f	---	Test MI 1
●	NVO	OLS 9	AI 9	SNVT_count_f	---	Test MI 2
●	NVI	OLA 6	AO 6	SNVT_count_f	---	Test MO 1
●	NVI	OLA 7	AO 7	SNVT_count_f	---	Test MO 2
●	NVO	OLS 10	AI 10	SNVT_count_f		Test MV 1

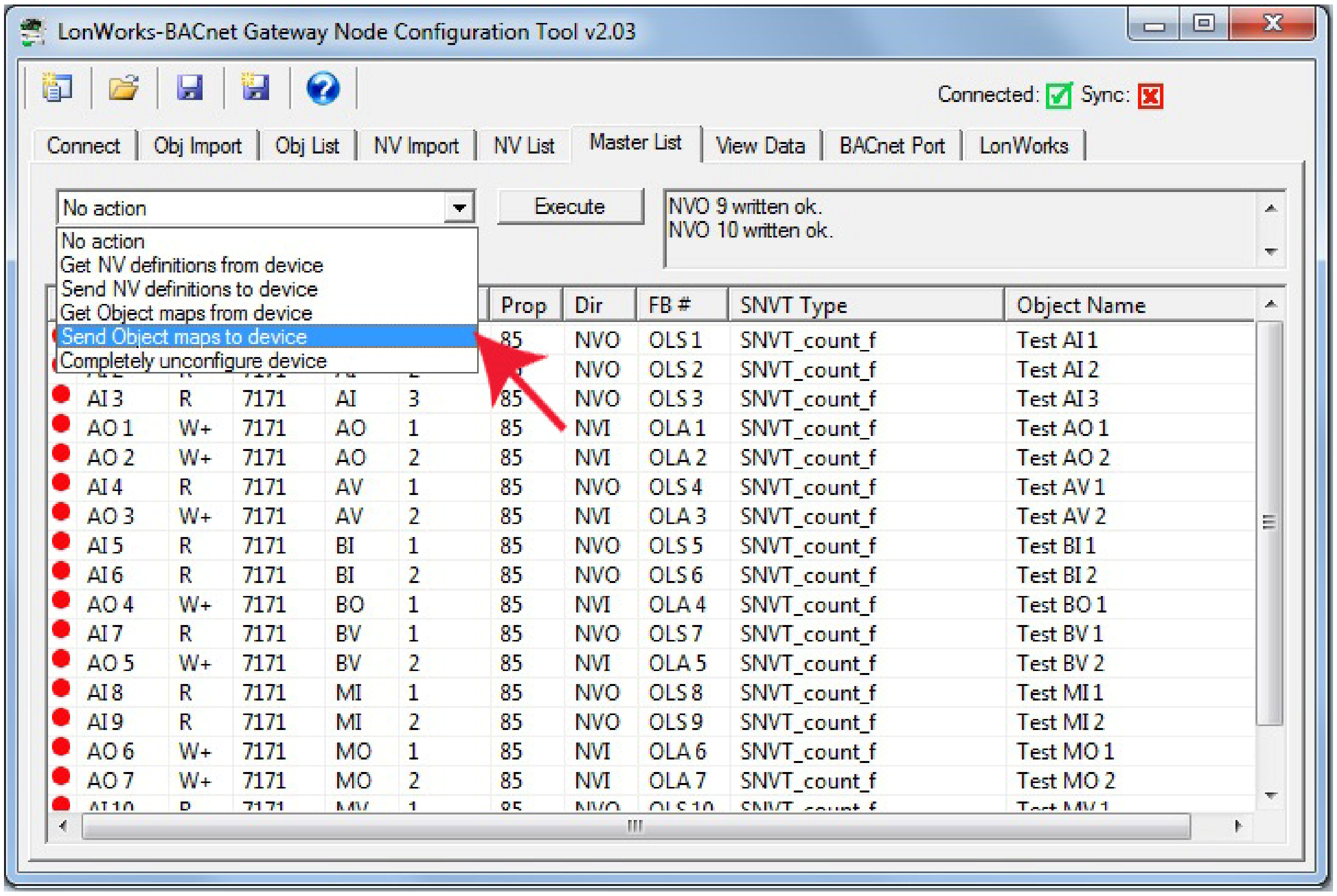
One of two steps have now been completed. Next, go to the Master List page. The Master List is, as the name implies, the master list for your device's configuration. This is where connections from LonWorks network variables to BACnet objects is made. Refer to the Master List section of this user guide for more detail about changes you can make from this page. For purposes of this example, we will assume that you are going to configure the gateway entirely with the default values that were automatically built.



```
NVO 9 written ok.
NVO 10 written ok.
```

	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	SNVT Type	Object Name
●	AI 1	R	7171	AI	1	85	NVO	OLS 1	SNVT_count_f	Test AI 1
●	AI 2	R	7171	AI	2	85	NVO	OLS 2	SNVT_count_f	Test AI 2
●	AI 3	R	7171	AI	3	85	NVO	OLS 3	SNVT_count_f	Test AI 3
●	AO 1	W+	7171	AO	1	85	NVI	OLA 1	SNVT_count_f	Test AO 1
●	AO 2	W+	7171	AO	2	85	NVI	OLA 2	SNVT_count_f	Test AO 2
●	AI 4	R	7171	AV	1	85	NVO	OLS 4	SNVT_count_f	Test AV 1
●	AO 3	W+	7171	AV	2	85	NVI	OLA 3	SNVT_count_f	Test AV 2
●	AI 5	R	7171	BI	1	85	NVO	OLS 5	SNVT_count_f	Test BI 1
●	AI 6	R	7171	BI	2	85	NVO	OLS 6	SNVT_count_f	Test BI 2
●	AO 4	W+	7171	BO	1	85	NVI	OLA 4	SNVT_count_f	Test BO 1
●	AI 7	R	7171	BV	1	85	NVO	OLS 7	SNVT_count_f	Test BV 1
●	AO 5	W+	7171	BV	2	85	NVI	OLA 5	SNVT_count_f	Test BV 2
●	AI 8	R	7171	MI	1	85	NVO	OLS 8	SNVT_count_f	Test MI 1
●	AI 9	R	7171	MI	2	85	NVO	OLS 9	SNVT_count_f	Test MI 2
●	AO 6	W+	7171	MO	1	85	NVI	OLA 6	SNVT_count_f	Test MO 1
●	AO 7	W+	7171	MO	2	85	NVI	OLA 7	SNVT_count_f	Test MO 2
●	AI 10	R	7171	MV	1	85	NVO	OLS 10	SNVT_count_f	Test MV 1

The second step of configuring the gateway device is to send the master list, or object maps, to the device. Select "Send Object maps to device" from the list and click Execute.

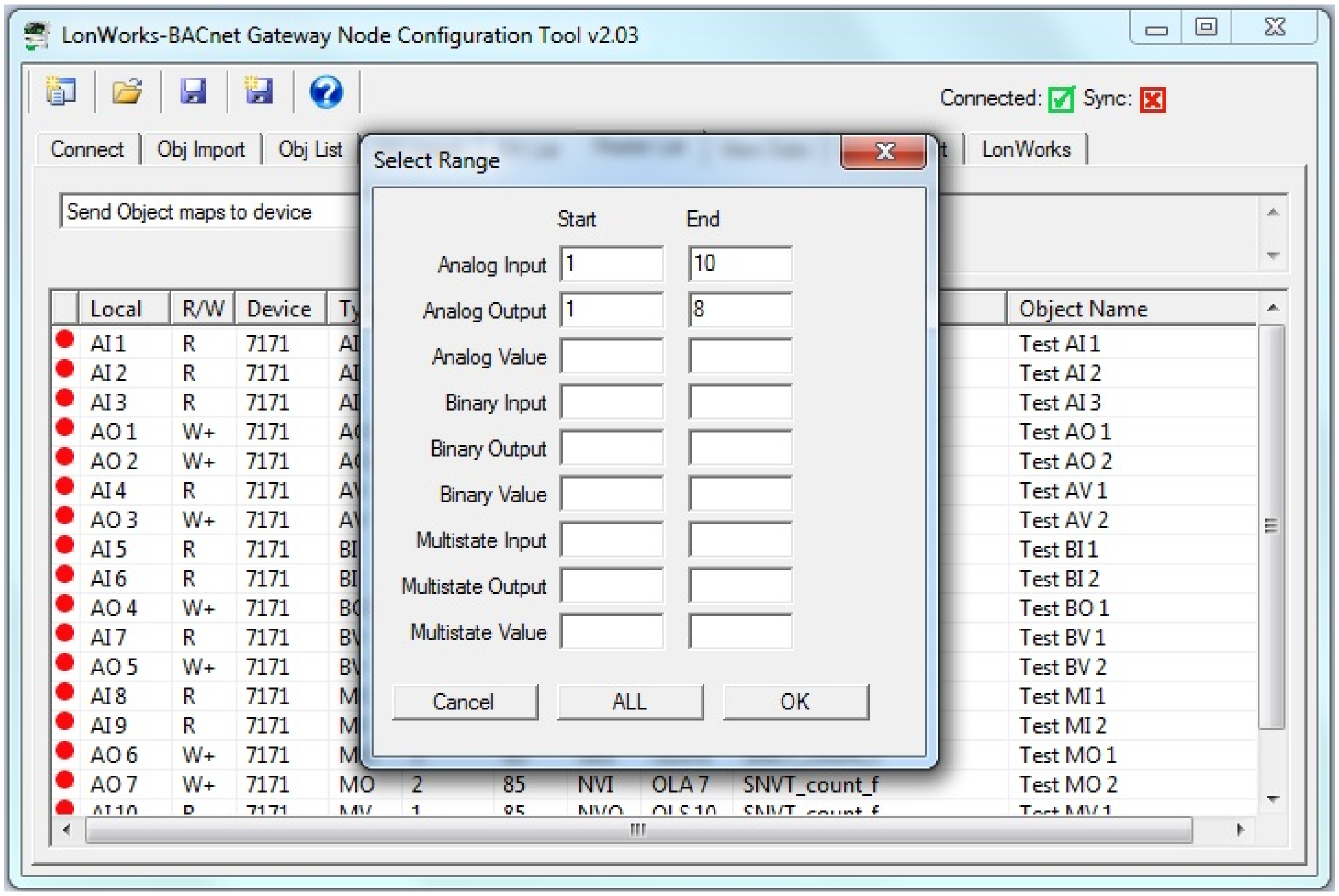

LonWorks-BACnet Gateway Node Configuration Tool v2.03
Connected:
Sync:
Connect
Obj Import
Obj List
NV Import
NV List
Master List
View Data
BACnet Port
LonWorks
Execute
No action
Get NV definitions from device
Send NV definitions to device
Get Object maps from device
Send Object maps to device
Completely unconfigure device

No action

NVO 9 written ok.
NVO 10 written ok.

						Prop	Dir	FB #	SNVT Type	Object Name
●						85	NVO	OLS 1	SNVT_count_f	Test AI 1
●	AI 2	R	7171	AI	2		NVO	OLS 2	SNVT_count_f	Test AI 2
●	AI 3	R	7171	AI	3	85	NVO	OLS 3	SNVT_count_f	Test AI 3
●	AO 1	W+	7171	AO	1	85	NVI	OLA 1	SNVT_count_f	Test AO 1
●	AO 2	W+	7171	AO	2	85	NVI	OLA 2	SNVT_count_f	Test AO 2
●	AI 4	R	7171	AV	1	85	NVO	OLS 4	SNVT_count_f	Test AV 1
●	AO 3	W+	7171	AV	2	85	NVI	OLA 3	SNVT_count_f	Test AV 2
●	AI 5	R	7171	BI	1	85	NVO	OLS 5	SNVT_count_f	Test BI 1
●	AI 6	R	7171	BI	2	85	NVO	OLS 6	SNVT_count_f	Test BI 2
●	AO 4	W+	7171	BO	1	85	NVI	OLA 4	SNVT_count_f	Test BO 1
●	AI 7	R	7171	BV	1	85	NVO	OLS 7	SNVT_count_f	Test BV 1
●	AO 5	W+	7171	BV	2	85	NVI	OLA 5	SNVT_count_f	Test BV 2
●	AI 8	R	7171	MI	1	85	NVO	OLS 8	SNVT_count_f	Test MI 1
●	AI 9	R	7171	MI	2	85	NVO	OLS 9	SNVT_count_f	Test MI 2
●	AO 6	W+	7171	MO	1	85	NVI	OLA 6	SNVT_count_f	Test MO 1
●	AO 7	W+	7171	MO	2	85	NVI	OLA 7	SNVT_count_f	Test MO 2
●	AI 10	R	7171	MV	1	85	NVO	OLS 10	SNVT_count_f	Test MV 1

Upon clicking Execute (with Send Objects to device selected), you will see the range dialog again. It will default to sending only those object maps that are valid but not yet sent to the device. Click OK to send the range of object maps shown.

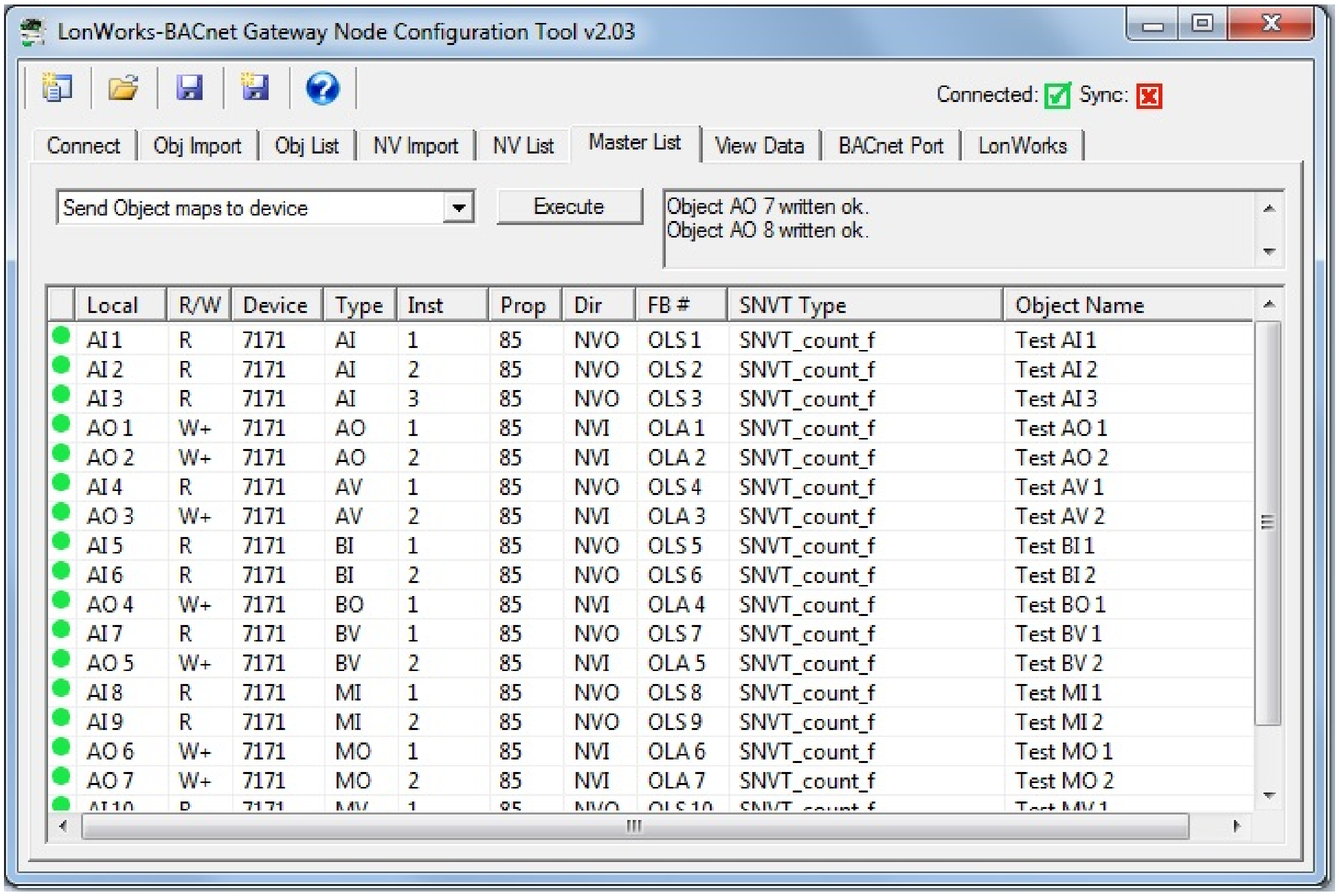

LonWorks-BACnet Gateway Node Configuration Tool v2.03
Connected: ✓ Sync: ✗
Connect
Obj Import
Obj List
LonWorks
Send Object maps to device
Select Range
Cancel
ALL
OK

	Start	End
Analog Input	1	10
Analog Output	1	8
Analog Value		
Binary Input		
Binary Output		
Binary Value		
Multistate Input		
Multistate Output		
Multistate Value		

	Local	R/W	Device	Ty						Object Name
●	AI 1	R	7171	AI						Test AI 1
●	AI 2	R	7171	AI						Test AI 2
●	AI 3	R	7171	AI						Test AI 3
●	AO 1	W+	7171	A						Test AO 1
●	AO 2	W+	7171	A						Test AO 2
●	AI 4	R	7171	A						Test AV 1
●	AO 3	W+	7171	A						Test AV 2
●	AI 5	R	7171	BI						Test BI 1
●	AI 6	R	7171	BI						Test BI 2
●	AO 4	W+	7171	B						Test BO 1
●	AI 7	R	7171	B						Test BV 1
●	AO 5	W+	7171	B						Test BV 2
●	AI 8	R	7171	M						Test MI 1
●	AI 9	R	7171	M						Test MI 2
●	AO 6	W+	7171	M						Test MO 1
●	AO 7	W+	7171	MO	2	85	NVI	OLA 7	SNVT_count_f	Test MO 2
●	AI 10	R	7171	MV	1	85	NVO	OLS 10	SNVT_count_f	Test MV 1

The icon in the first column will be red if the object map has not yet been sent to the gateway device. It will turn green after being sent to the device, and the configuration tool and the device being configured are now in sync.

If you close the tool software but are confident that the gateway device has already been configured, you can connect to the gateway and use "Get Object maps from device" to rebuild the list and confirm the configuration. When doing this, you should get NV definitions first, then get Object definitions. This is because the configuration will be incomplete if the object maps cannot find any network variables in the list to associate with.

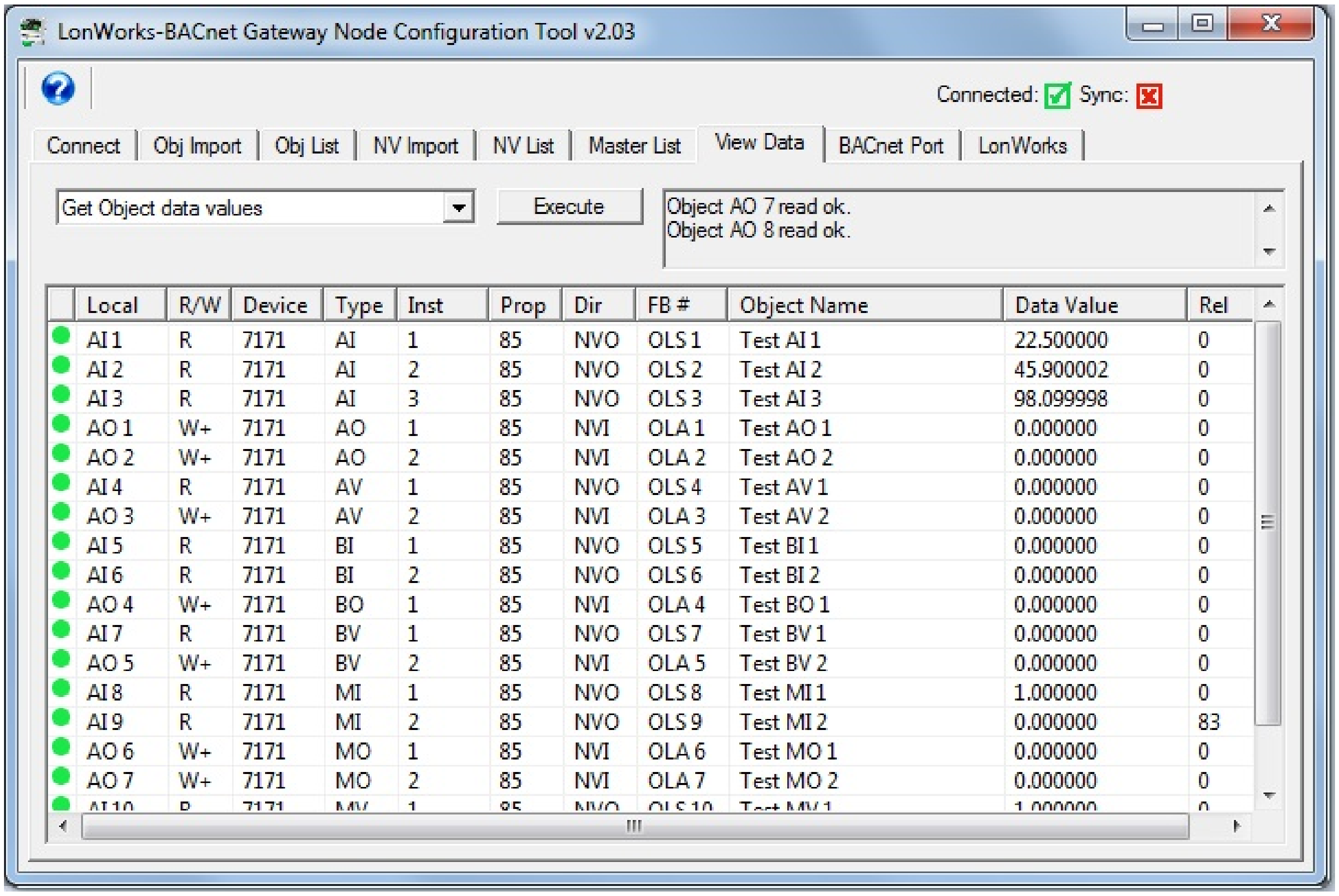


```
Object AO 7 written ok.
Object AO 8 written ok.
```

	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	SNVT Type	Object Name
●	AI 1	R	7171	AI	1	85	NVO	OLS 1	SNVT_count_f	Test AI 1
●	AI 2	R	7171	AI	2	85	NVO	OLS 2	SNVT_count_f	Test AI 2
●	AI 3	R	7171	AI	3	85	NVO	OLS 3	SNVT_count_f	Test AI 3
●	AO 1	W+	7171	AO	1	85	NVI	OLA 1	SNVT_count_f	Test AO 1
●	AO 2	W+	7171	AO	2	85	NVI	OLA 2	SNVT_count_f	Test AO 2
●	AI 4	R	7171	AV	1	85	NVO	OLS 4	SNVT_count_f	Test AV 1
●	AO 3	W+	7171	AV	2	85	NVI	OLA 3	SNVT_count_f	Test AV 2
●	AI 5	R	7171	BI	1	85	NVO	OLS 5	SNVT_count_f	Test BI 1
●	AI 6	R	7171	BI	2	85	NVO	OLS 6	SNVT_count_f	Test BI 2
●	AO 4	W+	7171	BO	1	85	NVI	OLA 4	SNVT_count_f	Test BO 1
●	AI 7	R	7171	BV	1	85	NVO	OLS 7	SNVT_count_f	Test BV 1
●	AO 5	W+	7171	BV	2	85	NVI	OLA 5	SNVT_count_f	Test BV 2
●	AI 8	R	7171	MI	1	85	NVO	OLS 8	SNVT_count_f	Test MI 1
●	AI 9	R	7171	MI	2	85	NVO	OLS 9	SNVT_count_f	Test MI 2
●	AO 6	W+	7171	MO	1	85	NVI	OLA 6	SNVT_count_f	Test MO 1
●	AO 7	W+	7171	MO	2	85	NVI	OLA 7	SNVT_count_f	Test MO 2
●	AI 10	R	7171	MV	1	85	NVO	OLS 10	SNVT_count_f	Test MV 1

The device configuration is largely complete once you have sent both the NV list and Object list to the device. You should also visit the BACnet Port page to set up things like MS/TP baud rate and MAC address, or BACnet IP address, etc. You may also want to visit the LonWorks page to set the node location property.

Once configuration is complete and the gateway is functioning as a gateway, you can retrieve a list of object data by going to the View Data page and selecting Get Object data values.



Object AO 7 read ok.
Object AO 8 read ok.

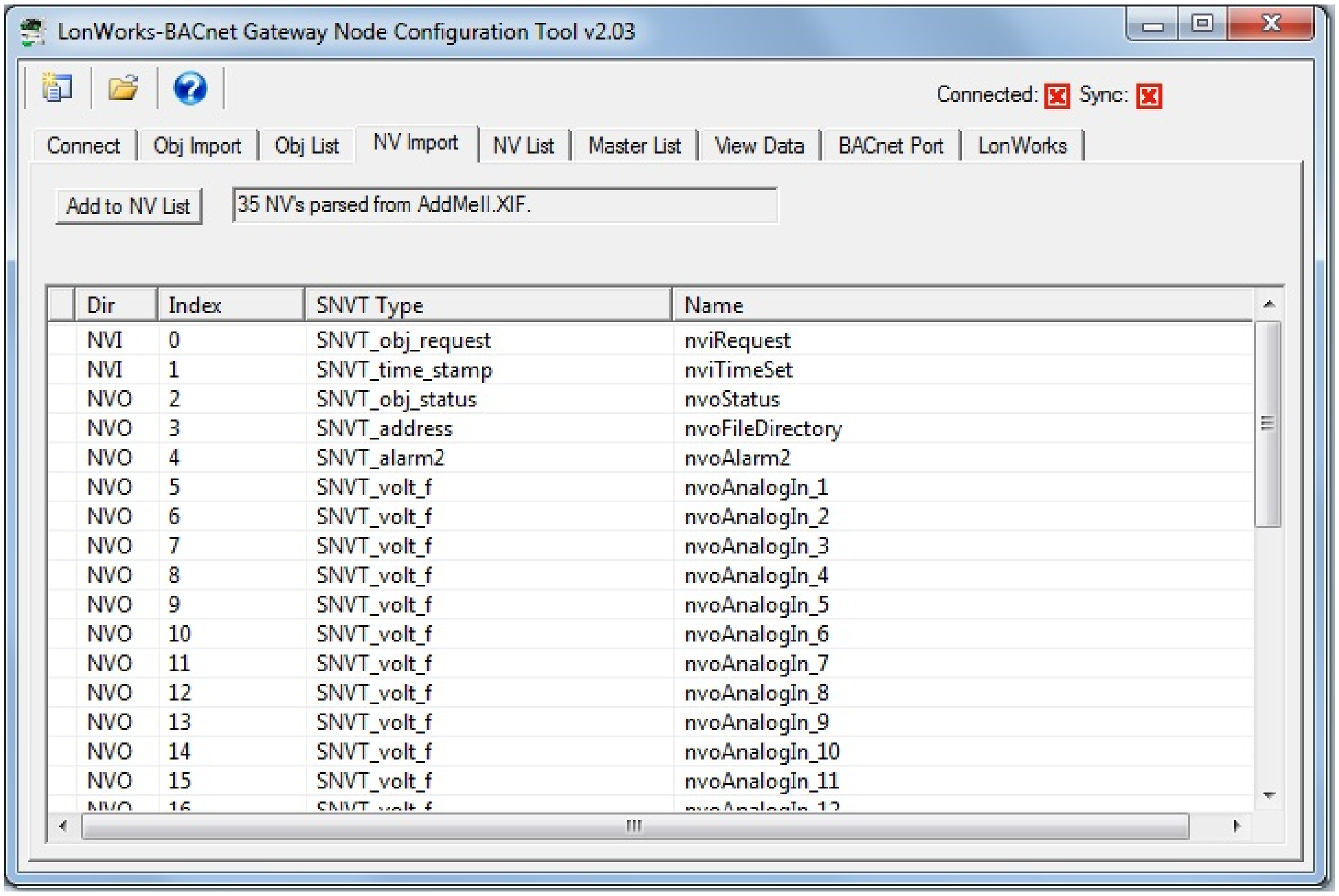
	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	Object Name	Data Value	Rel
●	AI 1	R	7171	AI	1	85	NVO	OLS 1	Test AI 1	22.500000	0
●	AI 2	R	7171	AI	2	85	NVO	OLS 2	Test AI 2	45.900002	0
●	AI 3	R	7171	AI	3	85	NVO	OLS 3	Test AI 3	98.099998	0
●	AO 1	W+	7171	AO	1	85	NVI	OLA 1	Test AO 1	0.000000	0
●	AO 2	W+	7171	AO	2	85	NVI	OLA 2	Test AO 2	0.000000	0
●	AI 4	R	7171	AV	1	85	NVO	OLS 4	Test AV 1	0.000000	0
●	AO 3	W+	7171	AV	2	85	NVI	OLA 3	Test AV 2	0.000000	0
●	AI 5	R	7171	BI	1	85	NVO	OLS 5	Test BI 1	0.000000	0
●	AI 6	R	7171	BI	2	85	NVO	OLS 6	Test BI 2	0.000000	0
●	AO 4	W+	7171	BO	1	85	NVI	OLA 4	Test BO 1	0.000000	0
●	AI 7	R	7171	BV	1	85	NVO	OLS 7	Test BV 1	0.000000	0
●	AO 5	W+	7171	BV	2	85	NVI	OLA 5	Test BV 2	0.000000	0
●	AI 8	R	7171	MI	1	85	NVO	OLS 8	Test MI 1	1.000000	0
●	AI 9	R	7171	MI	2	85	NVO	OLS 9	Test MI 2	0.000000	83
●	AO 6	W+	7171	MO	1	85	NVI	OLA 6	Test MO 1	0.000000	0
●	AO 7	W+	7171	MO	2	85	NVI	OLA 7	Test MO 2	0.000000	0
●	AI 10	R	7171	MV	1	85	NVO	OLS 10	Test MV 1	1.000000	0

3.3 Build Configuration from XIF File (LonWorks Device)

Starting from an XIF file for a specific LonWorks device, and auto-building the BACnet object list, is the approach you would take if you want to make a BACnet device mimic that particular LonWorks device on the LonWorks network.

IMPORTANT: If your intent is to put a LonWorks device on a BACnet network, this is not the correct gateway to use. The gateway discussed here only applies where you are interfacing BACnet devices to a LonWorks network, and the LonWorks network is the primary network of the overall system.

To begin building your configuration from an XIF file, skip the Obj Import page and go to the NV Import page. Click on the file icon to open an XIF file. Once the XIF is imported, the list of network variables will be displayed on the NV Import page.

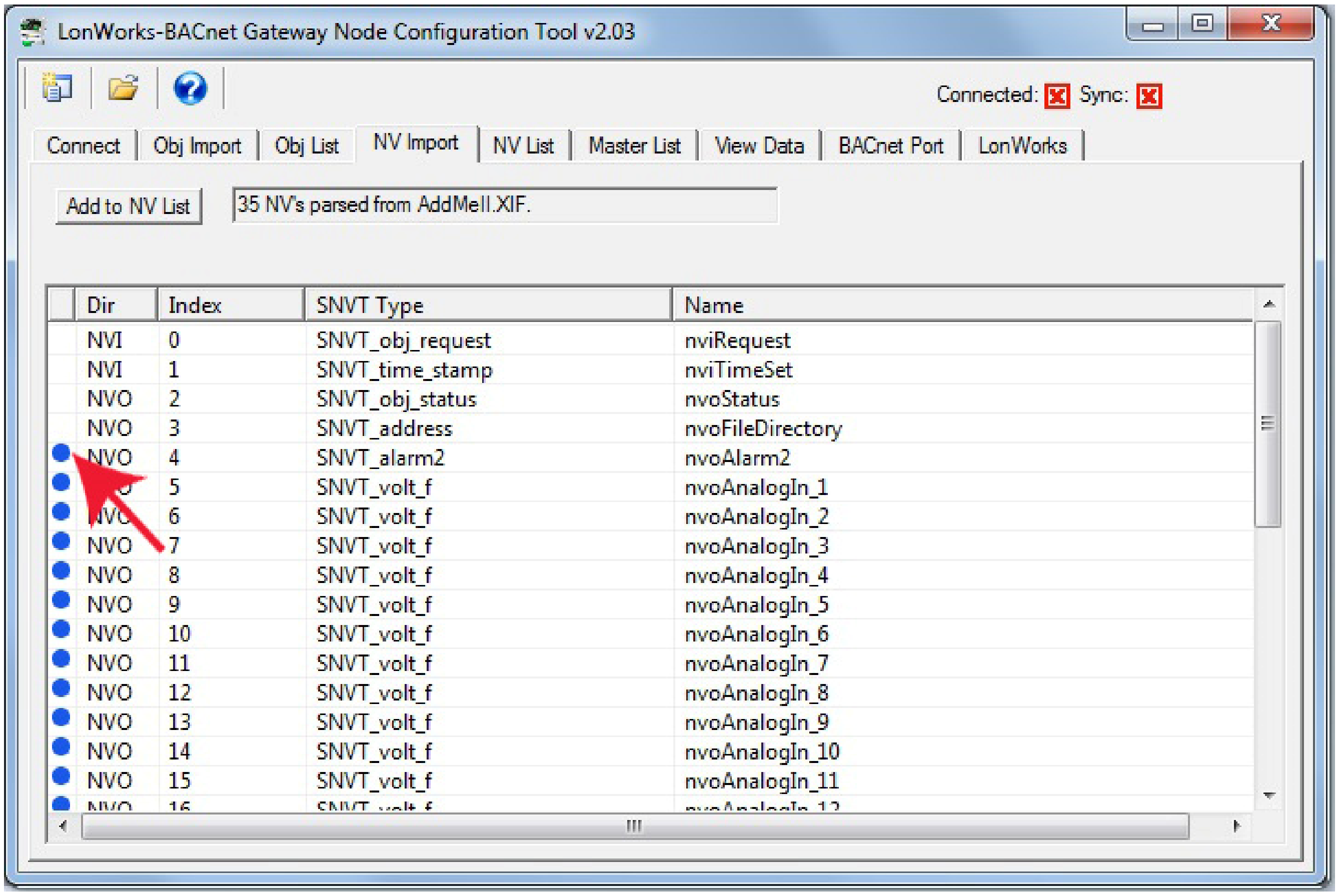


35 NV's parsed from AddMeII.XIF.

Dir	Index	SNVT Type	Name
NVI	0	SNVT_obj_request	nviRequest
NVI	1	SNVT_time_stamp	nviTimeSet
NVO	2	SNVT_obj_status	nvoStatus
NVO	3	SNVT_address	nvoFileDirectory
NVO	4	SNVT_alarm2	nvoAlarm2
NVO	5	SNVT_volt_f	nvoAnalogIn_1
NVO	6	SNVT_volt_f	nvoAnalogIn_2
NVO	7	SNVT_volt_f	nvoAnalogIn_3
NVO	8	SNVT_volt_f	nvoAnalogIn_4
NVO	9	SNVT_volt_f	nvoAnalogIn_5
NVO	10	SNVT_volt_f	nvoAnalogIn_6
NVO	11	SNVT_volt_f	nvoAnalogIn_7
NVO	12	SNVT_volt_f	nvoAnalogIn_8
NVO	13	SNVT_volt_f	nvoAnalogIn_9
NVO	14	SNVT_volt_f	nvoAnalogIn_10
NVO	15	SNVT_volt_f	nvoAnalogIn_11
NVO	16	SNVT_volt_f	nvoAnalogIn_12

The NV Import page is essentially a scratch pad where you import the content of XIF files, then select which of the available variables you wish to include in your gateway configuration. Click on the icon column header to select all items, or click on the icon column for individual lines to select only those lines. The icon will show a blue dot for those lines that are about to be included.

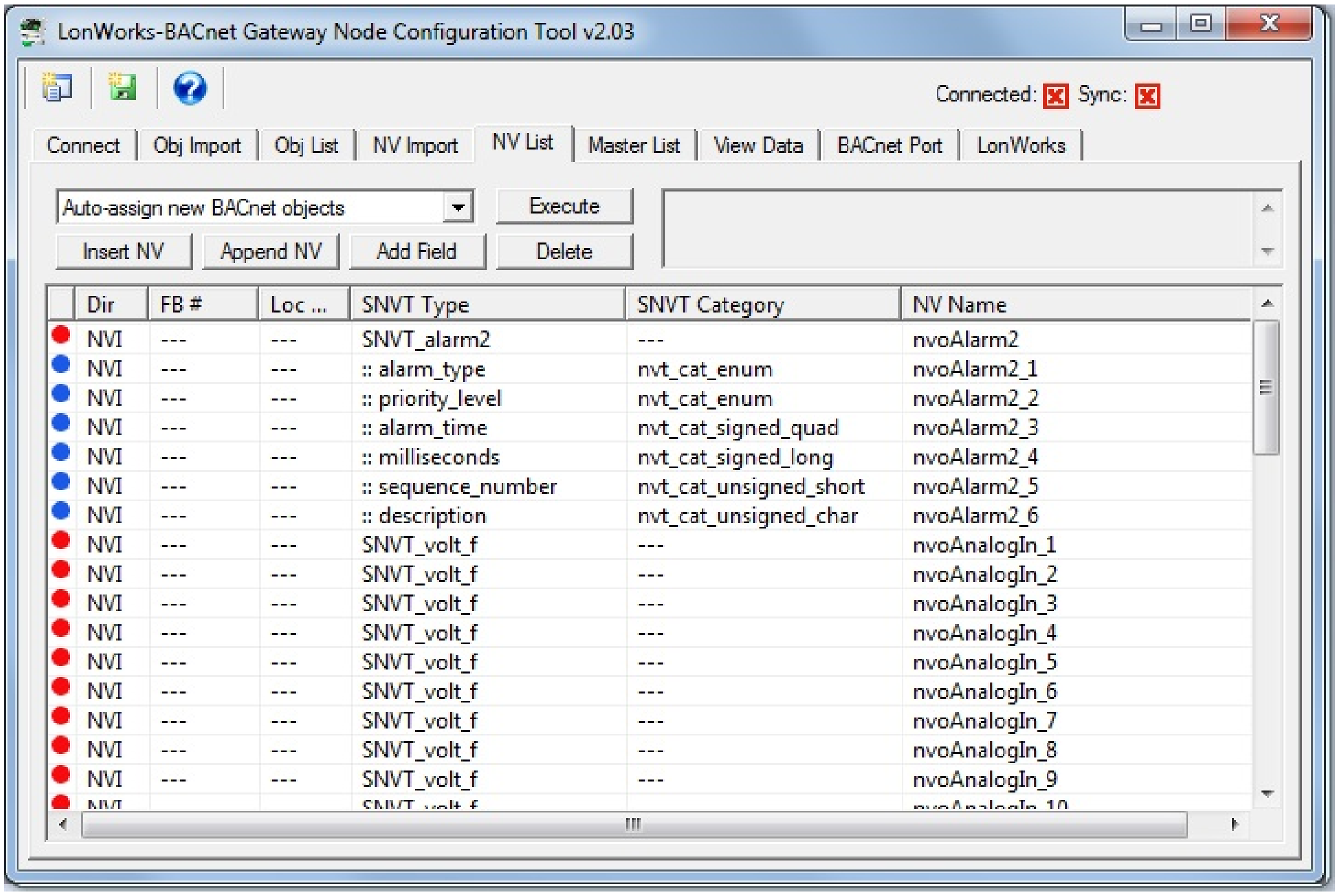
When you have made your selections, click Add to NV List. The selected variables are now copied to the NV List.



35 NV's parsed from AddMeII.XIF.

	Dir	Index	SNVT Type	Name
	NVI	0	SNVT_obj_request	nviRequest
	NVI	1	SNVT_time_stamp	nviTimeSet
	NVO	2	SNVT_obj_status	nvoStatus
	NVO	3	SNVT_address	nvoFileDirectory
●	NVO	4	SNVT_alarm2	nvoAlarm2
●	NVO	5	SNVT_volt_f	nvoAnalogIn_1
●	NVO	6	SNVT_volt_f	nvoAnalogIn_2
●	NVO	7	SNVT_volt_f	nvoAnalogIn_3
●	NVO	8	SNVT_volt_f	nvoAnalogIn_4
●	NVO	9	SNVT_volt_f	nvoAnalogIn_5
●	NVO	10	SNVT_volt_f	nvoAnalogIn_6
●	NVO	11	SNVT_volt_f	nvoAnalogIn_7
●	NVO	12	SNVT_volt_f	nvoAnalogIn_8
●	NVO	13	SNVT_volt_f	nvoAnalogIn_9
●	NVO	14	SNVT_volt_f	nvoAnalogIn_10
●	NVO	15	SNVT_volt_f	nvoAnalogIn_11
●	NVO	16	SNVT_volt_f	nvoAnalogIn_12

The NV List is the definition of the list of network variables that will become present in your LonWorks device (i.e. this gateway) when viewed as a node on the LonWorks network. The icon in the first column will be red if this NV definition has not yet been written to the gateway, and green if it has been written to the gateway. Blue icons indicate fields of structured network variables. All lines with a blue icon are part of the network variable immediately preceding the set of blue icons. There will be only one network variable, but multiple BACnet objects, for a structured network variable (refer to previous section in this user guide if you did not already review treatment of structured variables).

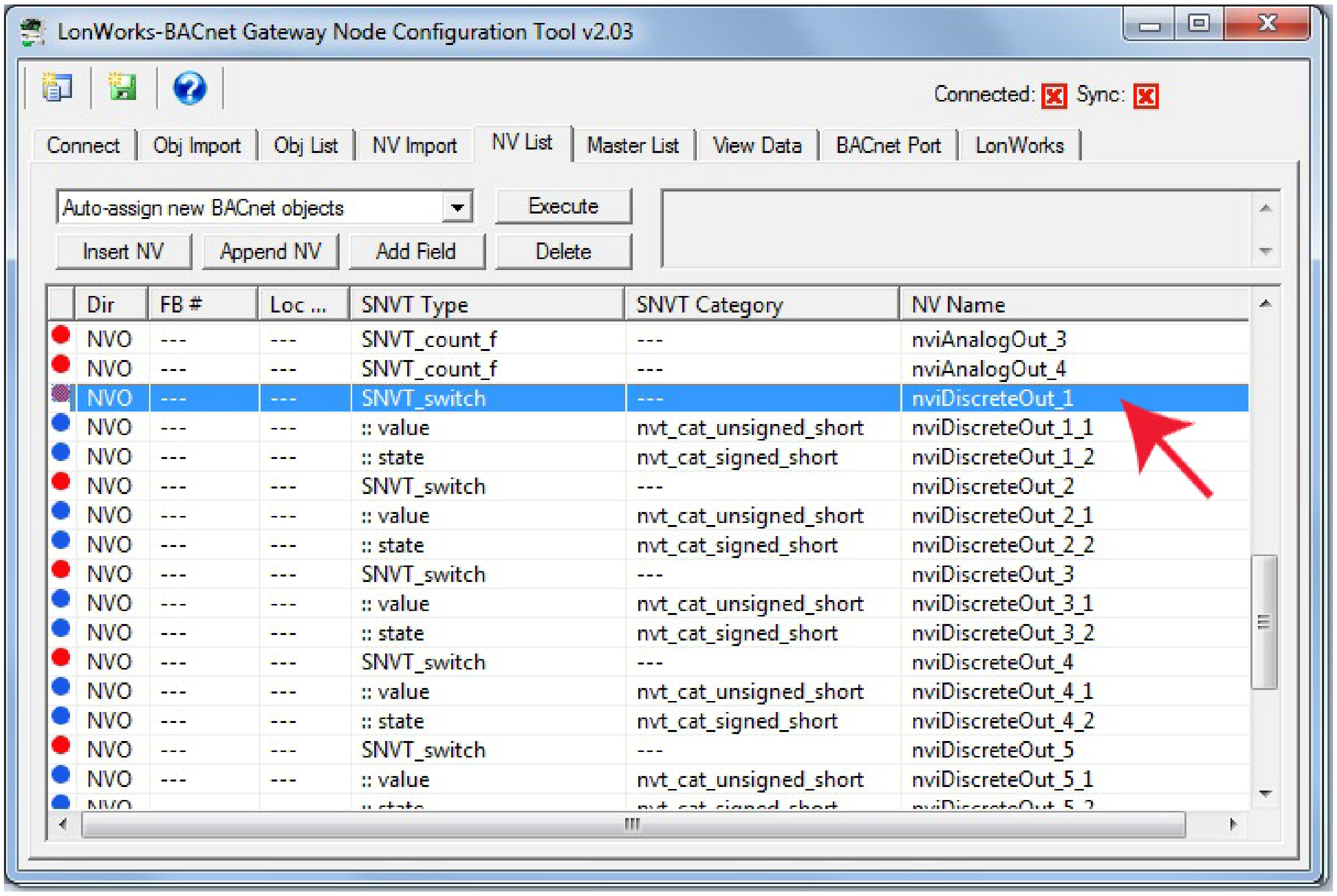


	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVI	---	---	SNVT_alarm2	---	nvoAlarm2
●	NVI	---	---	:: alarm_type	nvt_cat_enum	nvoAlarm2_1
●	NVI	---	---	:: priority_level	nvt_cat_enum	nvoAlarm2_2
●	NVI	---	---	:: alarm_time	nvt_cat_signed_quad	nvoAlarm2_3
●	NVI	---	---	:: milliseconds	nvt_cat_signed_long	nvoAlarm2_4
●	NVI	---	---	:: sequence_number	nvt_cat_unsigned_short	nvoAlarm2_5
●	NVI	---	---	:: description	nvt_cat_unsigned_char	nvoAlarm2_6
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_1
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_2
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_3
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_4
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_5
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_6
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_7
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_8
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_9
●	NVI			SNVT_volt_f		nvoAnalogIn_10

If you wish to make changes to the NV List, this is the point where you should do so. Do not proceed to assign function blocks (FB #'s) or data objects until the NV List is finalized.

You have a few options here, and these are described in more detail in the NV List section of this user guide. You may add network variables, delete them, or change what type they are. If they are structured, they will be automatically expanded into a list of all of their fields. If you add a structured NV which is not a standard LonMark type, you will need to add fields manually to build up the structure.

Some network variable types provide for special conversions. SNVT_switch is one such NV. To modify the NV definition, double click on the respective line in the NV List.



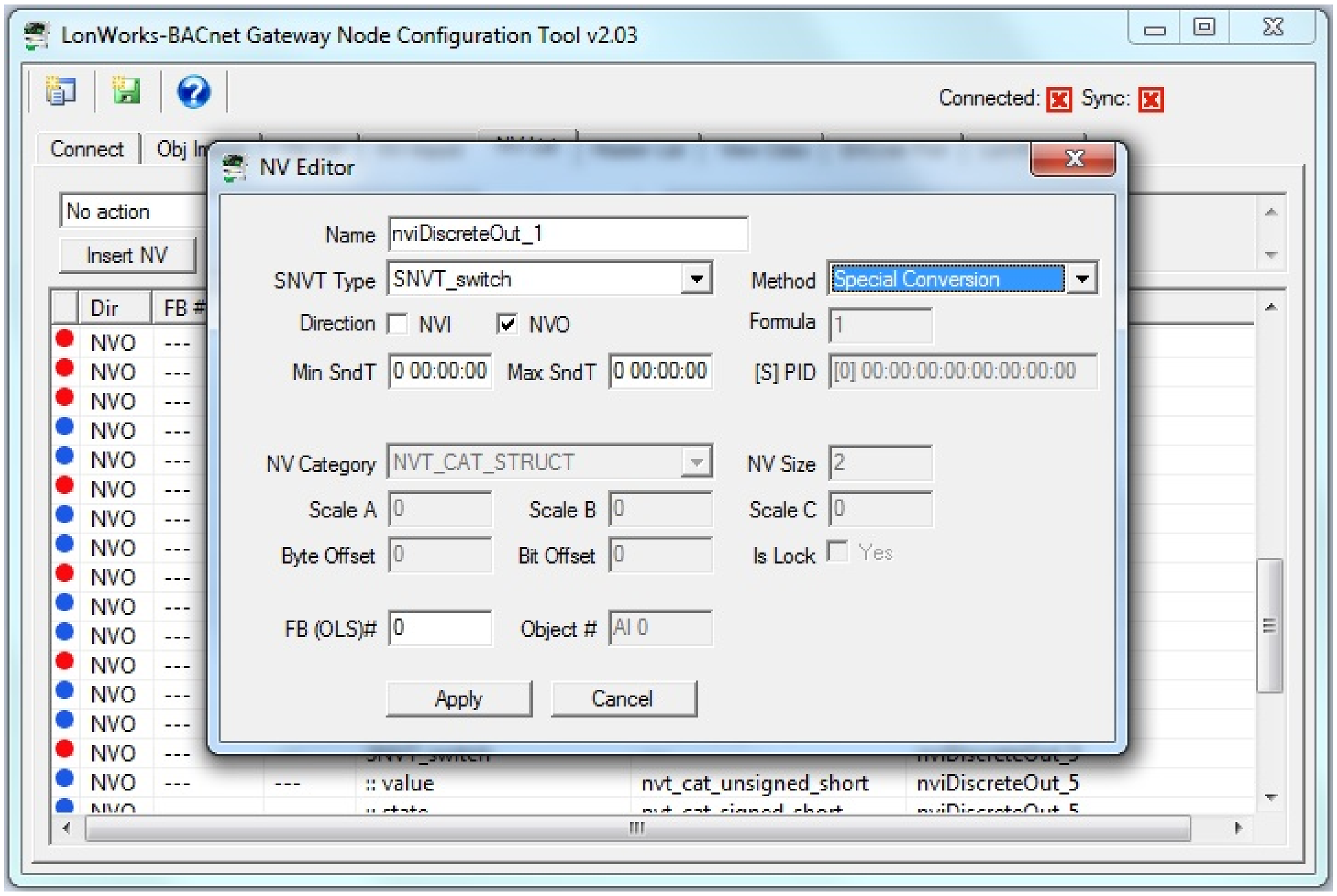
Auto-assign new BACnet objects

	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVO	---	---	SNVT_count_f	---	nviAnalogOut_3
●	NVO	---	---	SNVT_count_f	---	nviAnalogOut_4
●	NVO	---	---	SNVT_switch	---	nviDiscreteOut_1
●	NVO	---	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_1_1
●	NVO	---	---	:: state	nvt_cat_signed_short	nviDiscreteOut_1_2
●	NVO	---	---	SNVT_switch	---	nviDiscreteOut_2
●	NVO	---	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_2_1
●	NVO	---	---	:: state	nvt_cat_signed_short	nviDiscreteOut_2_2
●	NVO	---	---	SNVT_switch	---	nviDiscreteOut_3
●	NVO	---	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_3_1
●	NVO	---	---	:: state	nvt_cat_signed_short	nviDiscreteOut_3_2
●	NVO	---	---	SNVT_switch	---	nviDiscreteOut_4
●	NVO	---	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_4_1
●	NVO	---	---	:: state	nvt_cat_signed_short	nviDiscreteOut_4_2
●	NVO	---	---	SNVT_switch	---	nviDiscreteOut_5
●	NVO	---	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_5_1
●	NVO			:: state	nvt_cat_signed_short	nviDiscreteOut_5_2

Upon double clicking a network variable in the NV List, the NV Editor dialog will appear. The SNVT_switch example is illustrated here. The LonMark definition of a "switch" actually contains two elements of data, a state and a level (as would be used for a dimmer switch). You cannot correctly control anything via a SNVT_switch without properly dealing with both elements of data. In BACnet terms, this would require two BACnet objects to control one switch, and this is not typically desirable from a BACnet point of view. Therefore, the Babel Buster gateway provides a "special conversion" such that a single BACnet object containing a value from 0 to 100 (implied percent) will result in both parts of the SNVT_switch being set correctly to control on/off (100% or 0%) or any level in between.

Your selection of whether to use the special conversion will decide whether SNVT_switch maps to one or two BACnet objects. For this reason, it is important that you make all of your NV related selections and configurations BEFORE assigning function blocks and data objects.

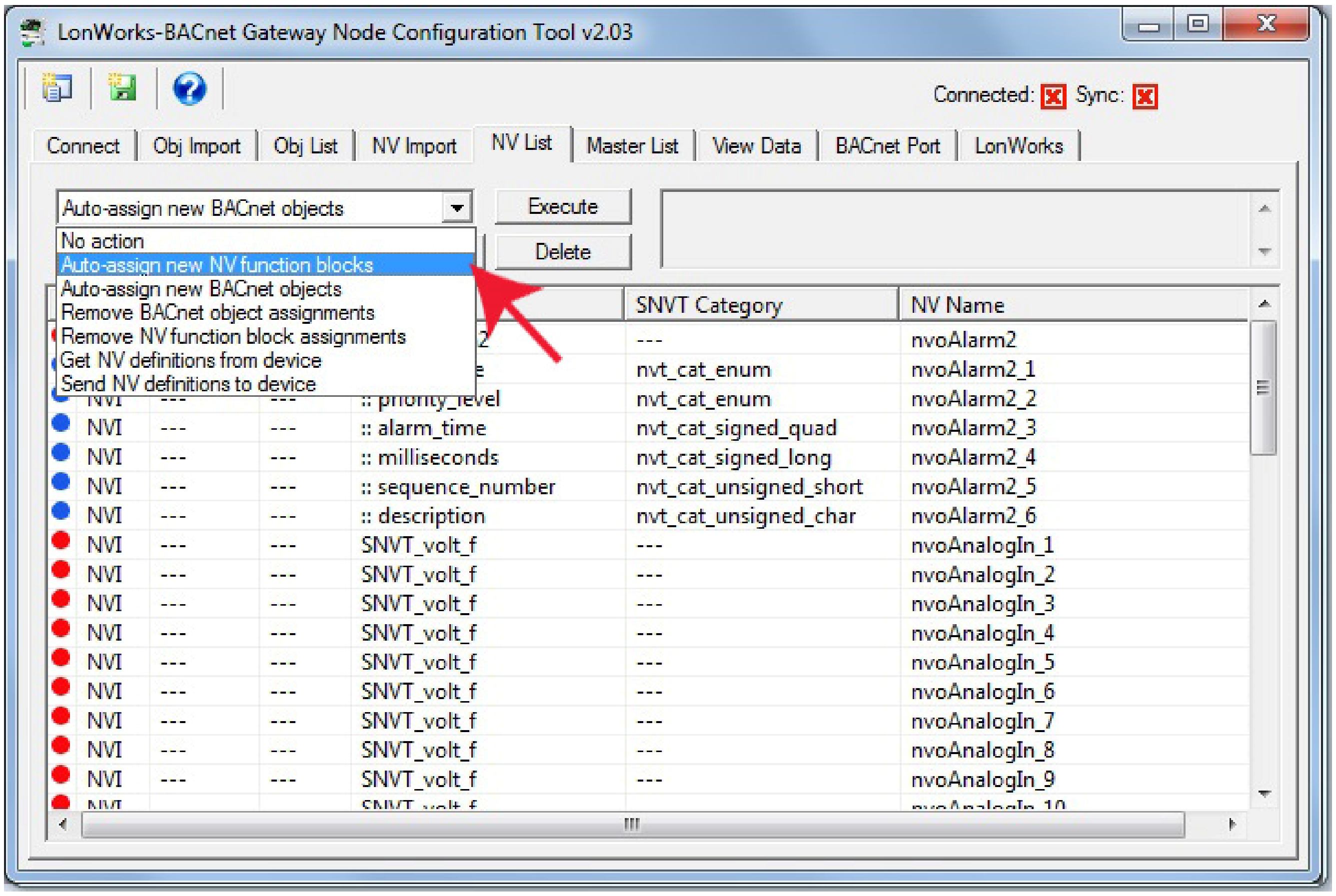
You will also be given the option of automatically converting all instances of SNVT_switch to single objects during the auto-assignment process noted below.



Name: nviDiscreteOut_1
SNVT Type: SNVT_switch
Method: Special Conversion
Direction: ☐ NVI ☑ NVO
Formula: 1
Min SndT: 0 00:00:00
Max SndT: 0 00:00:00
[S] PID: [0] 00:00:00:00:00:00:00:00
NV Category: NVT_CAT_STRUCT
NV Size: 2
Scale A: 0
Scale B: 0
Scale C: 0
Byte Offset: 0
Bit Offset: 0
Is Lock: ☐ Yes
FB (OLS)#: 0
Object #: AI 0

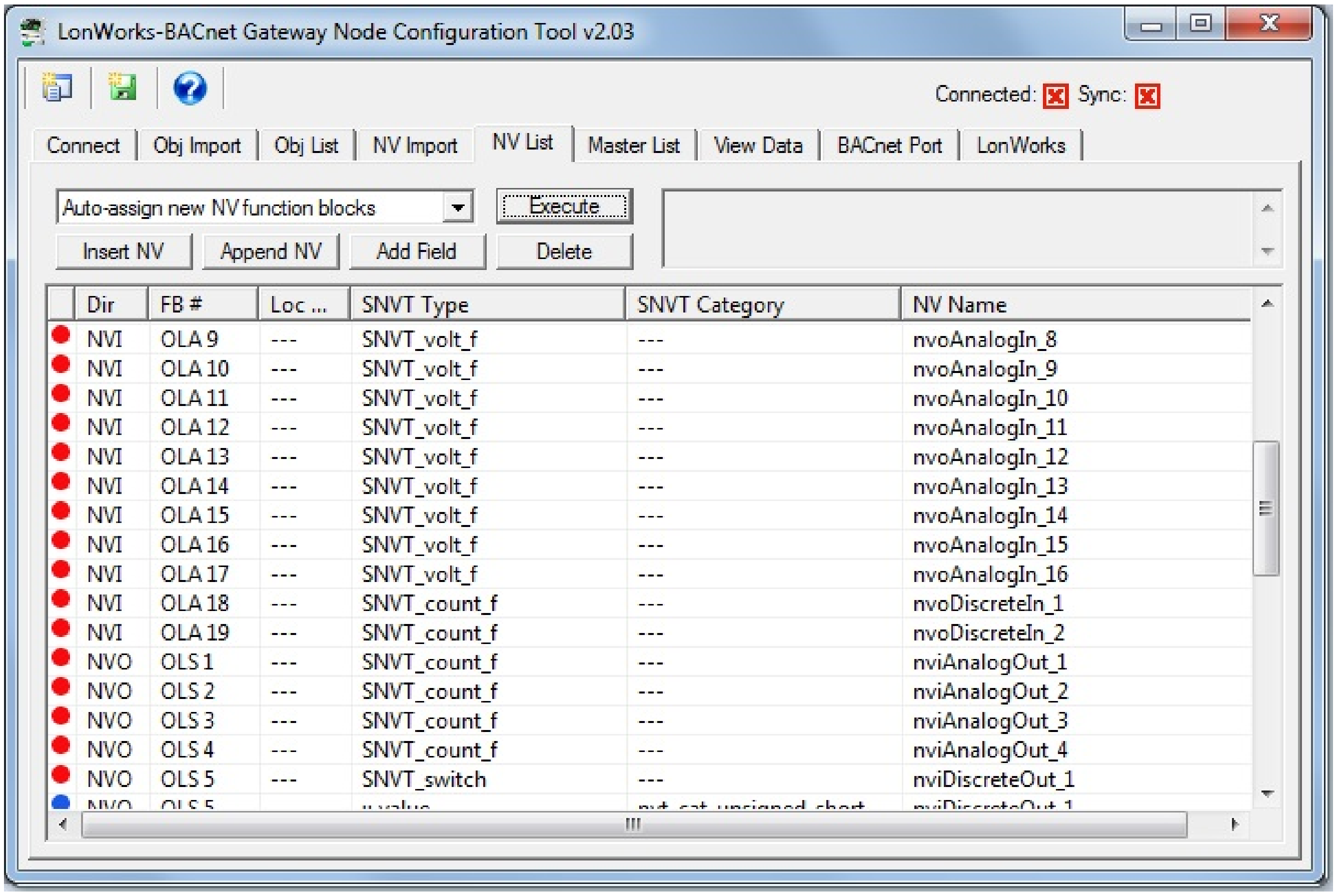
Dir	FB #				
NVO	---				
NVO	---				
NVO	---				
NVO	---				
NVO	---				
NVO	---				
NVO	---				
NVO	---				
NVO	---				
NVO	---				
NVO	---				
NVO	---				
NVO	---				
NVO	---				
NVO	---				
NVO	---	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_5
NVO			:: state	nvt_cat_signed_short	nviDiscreteOut_5

Once you have made all of the desired modifications to network variables in the NV List, proceed to assign function blocks. Select "Auto-assign new NV function blocks" and click Execute.


LonWorks-BACnet Gateway Node Configuration Tool v2.03
Connected: Sync:
Connect
Obj Import
Obj List
NV Import
NV List
Master List
View Data
BACnet Port
LonWorks
Auto-assign new BACnet objects
Execute
Delete
No action
Auto-assign new NV function blocks
Auto-assign new BACnet objects
Remove BACnet object assignments
Remove NV function block assignments
Get NV definitions from device
Send NV definitions to device

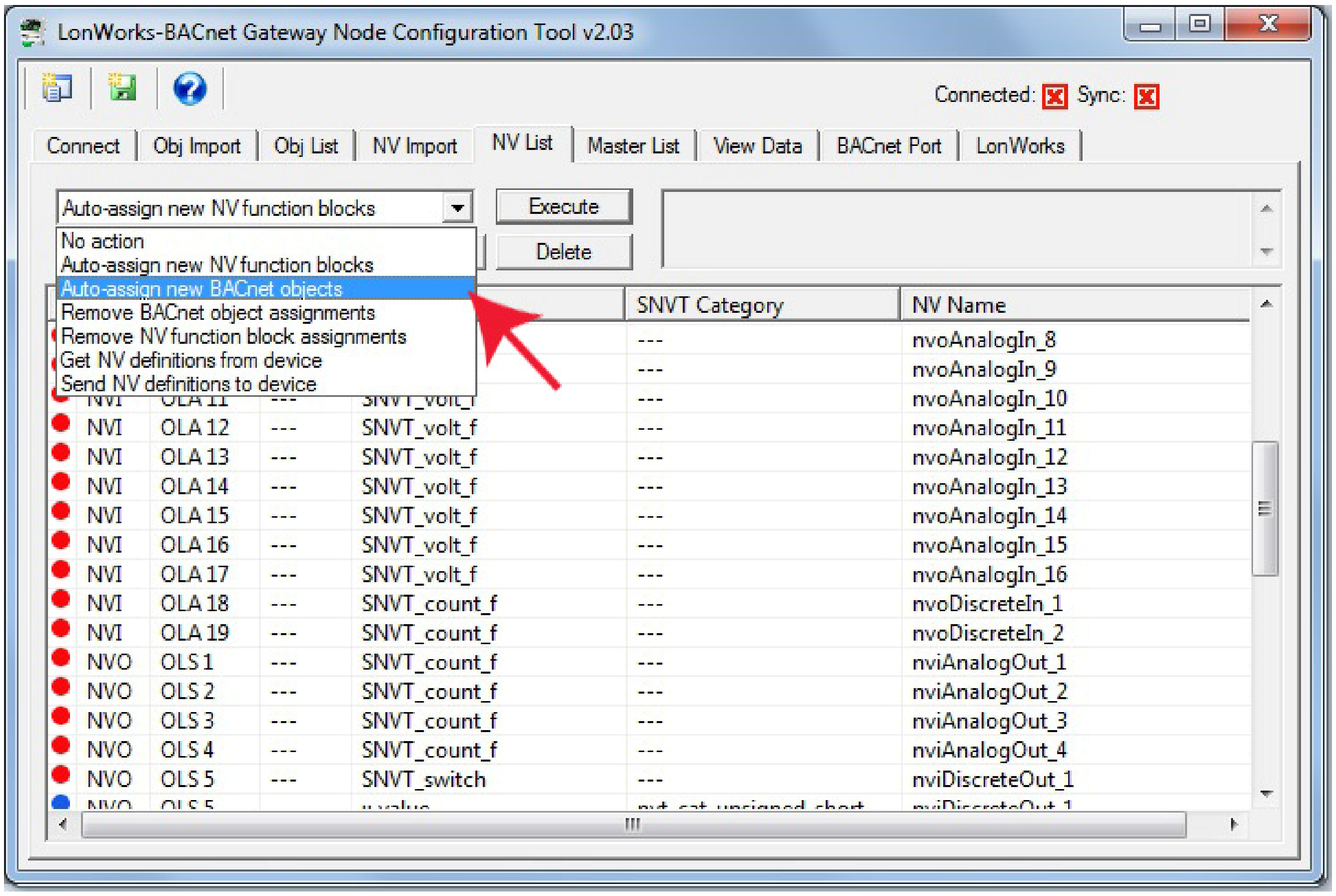
					SNVT Category	NV Name
					---	nvoAlarm2
					nvt_cat_enum	nvoAlarm2_1
	NVI	---	---	:: priority_level	nvt_cat_enum	nvoAlarm2_2
●	NVI	---	---	:: alarm_time	nvt_cat_signed_quad	nvoAlarm2_3
●	NVI	---	---	:: milliseconds	nvt_cat_signed_long	nvoAlarm2_4
●	NVI	---	---	:: sequence_number	nvt_cat_unsigned_short	nvoAlarm2_5
●	NVI	---	---	:: description	nvt_cat_unsigned_char	nvoAlarm2_6
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_1
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_2
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_3
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_4
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_5
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_6
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_7
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_8
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_9
●	NVI			SNVT_volt_f		nvoAnalogIn_10

Upon execution of the function block assignments, the FB# column will be populated.



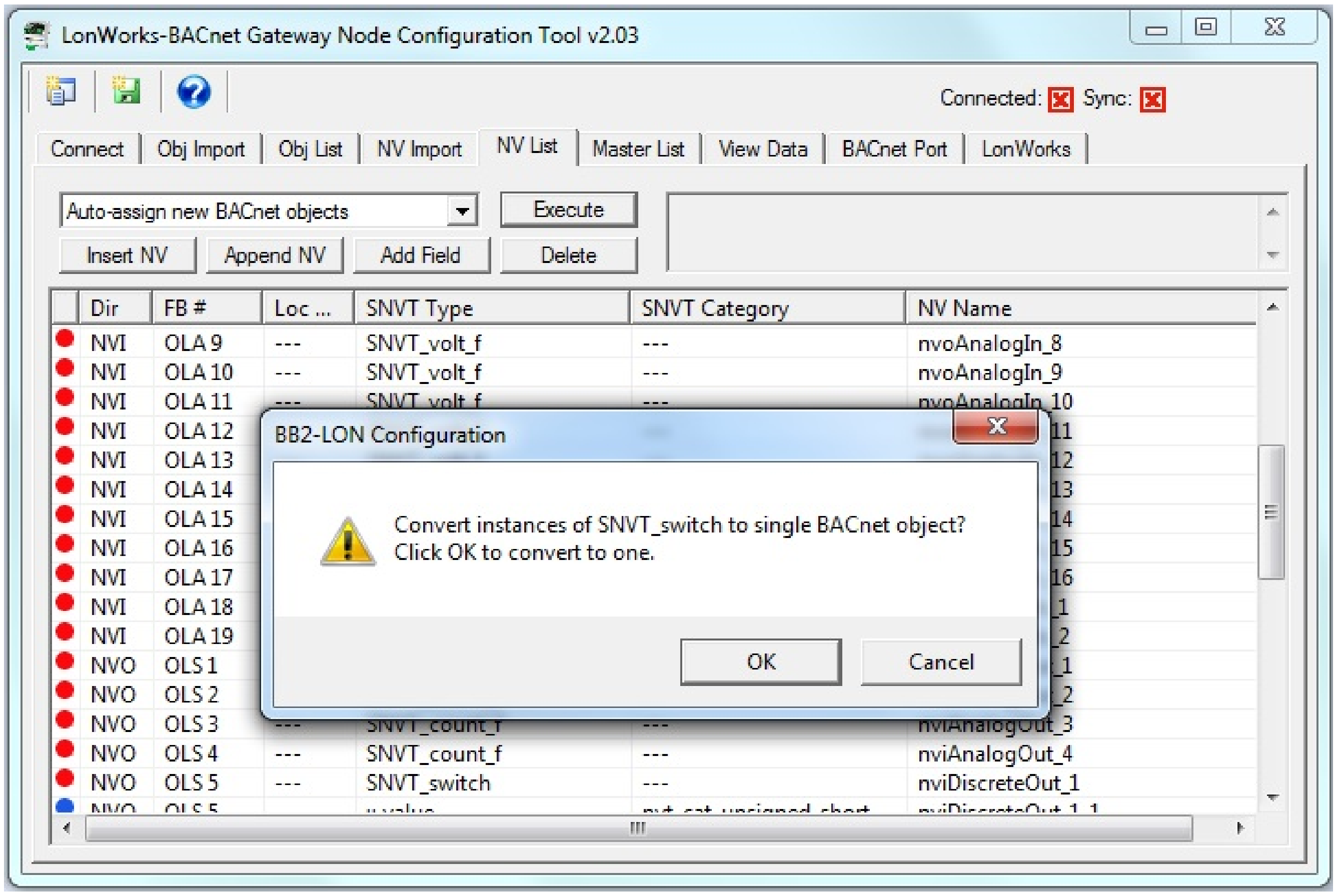
	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVI	OLA 9	---	SNVT_volt_f	---	nvoAnalogIn_8
●	NVI	OLA 10	---	SNVT_volt_f	---	nvoAnalogIn_9
●	NVI	OLA 11	---	SNVT_volt_f	---	nvoAnalogIn_10
●	NVI	OLA 12	---	SNVT_volt_f	---	nvoAnalogIn_11
●	NVI	OLA 13	---	SNVT_volt_f	---	nvoAnalogIn_12
●	NVI	OLA 14	---	SNVT_volt_f	---	nvoAnalogIn_13
●	NVI	OLA 15	---	SNVT_volt_f	---	nvoAnalogIn_14
●	NVI	OLA 16	---	SNVT_volt_f	---	nvoAnalogIn_15
●	NVI	OLA 17	---	SNVT_volt_f	---	nvoAnalogIn_16
●	NVI	OLA 18	---	SNVT_count_f	---	nvoDiscreteIn_1
●	NVI	OLA 19	---	SNVT_count_f	---	nvoDiscreteIn_2
●	NVO	OLS 1	---	SNVT_count_f	---	nviAnalogOut_1
●	NVO	OLS 2	---	SNVT_count_f	---	nviAnalogOut_2
●	NVO	OLS 3	---	SNVT_count_f	---	nviAnalogOut_3
●	NVO	OLS 4	---	SNVT_count_f	---	nviAnalogOut_4
●	NVO	OLS 5	---	SNVT_switch	---	nviDiscreteOut_1
●	NVO	OLS 5		u.value	nvt_cat_unsigned_short	nviDiscreteOut_1

Next, select and execute "Auto-assign new BACnet objects".


LonWorks-BACnet Gateway Node Configuration Tool v2.03
Connected:
Sync:
Connect
Obj Import
Obj List
NV Import
NV List
Master List
View Data
BACnet Port
LonWorks
Auto-assign new NV function blocks
Execute
Delete
No action
Auto-assign new NV function blocks
Auto-assign new BACnet objects
Remove BACnet object assignments
Remove NV function block assignments
Get NV definitions from device
Send NV definitions to device

					SNVT Category	NV Name
					---	nvoAnalogIn_8
					---	nvoAnalogIn_9
●	NVI	OLA 11	---	SNVT_volt_f	---	nvoAnalogIn_10
●	NVI	OLA 12	---	SNVT_volt_f	---	nvoAnalogIn_11
●	NVI	OLA 13	---	SNVT_volt_f	---	nvoAnalogIn_12
●	NVI	OLA 14	---	SNVT_volt_f	---	nvoAnalogIn_13
●	NVI	OLA 15	---	SNVT_volt_f	---	nvoAnalogIn_14
●	NVI	OLA 16	---	SNVT_volt_f	---	nvoAnalogIn_15
●	NVI	OLA 17	---	SNVT_volt_f	---	nvoAnalogIn_16
●	NVI	OLA 18	---	SNVT_count_f	---	nvoDiscreteIn_1
●	NVI	OLA 19	---	SNVT_count_f	---	nvoDiscreteIn_2
●	NVO	OLS 1	---	SNVT_count_f	---	nviAnalogOut_1
●	NVO	OLS 2	---	SNVT_count_f	---	nviAnalogOut_2
●	NVO	OLS 3	---	SNVT_count_f	---	nviAnalogOut_3
●	NVO	OLS 4	---	SNVT_count_f	---	nviAnalogOut_4
●	NVO	OLS 5	---	SNVT_switch	---	nviDiscreteOut_1
●	NVO	OLS 5		u.value	nvt_cat_unsigned_short	nviDiscreteOut_1

Upon clicking Execute, you will be given the option of automatically converting all instances of SNVT_switch to single objects. This is a short cut alternative to the manual editing of SNVT_switch variables noted above.

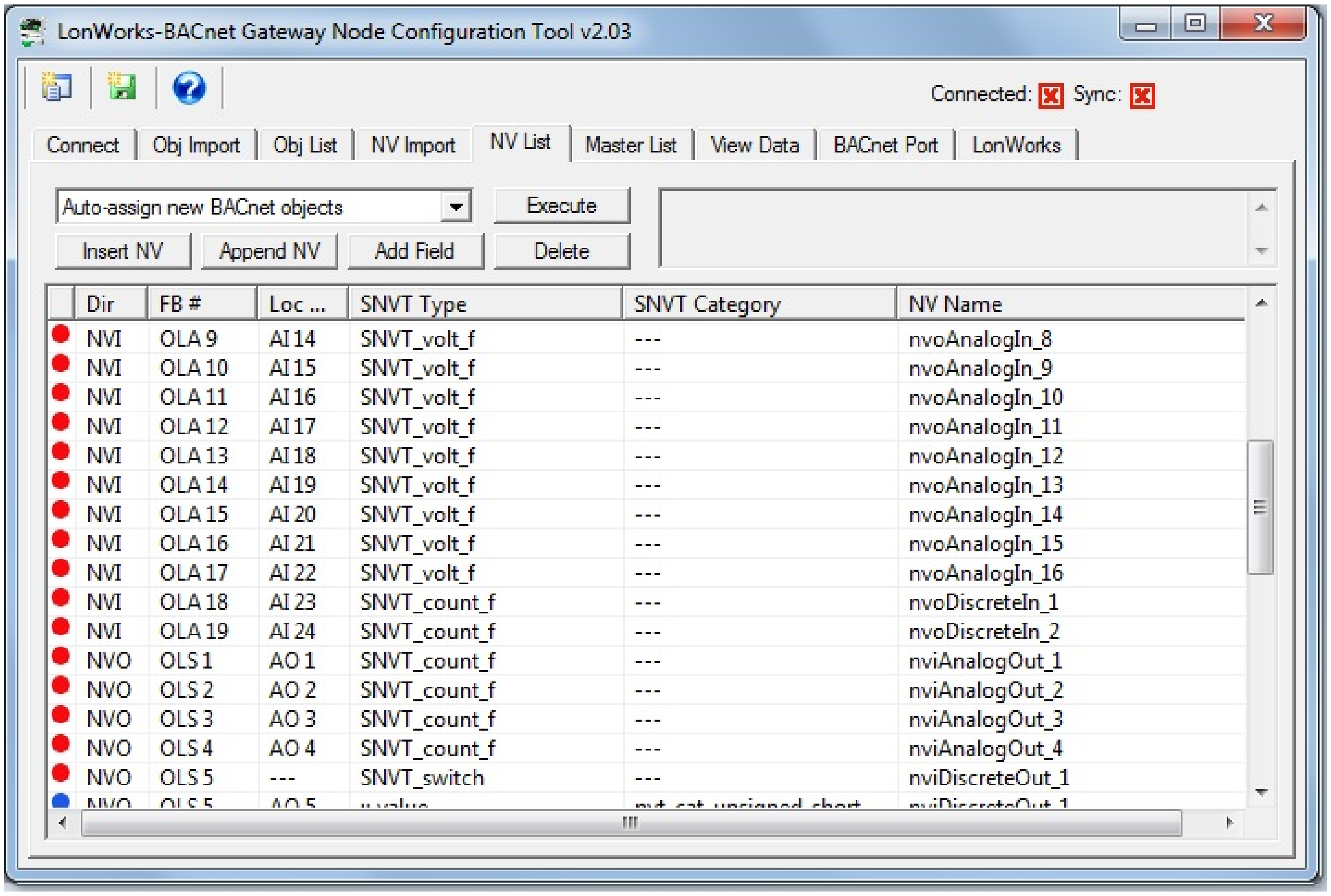


	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVI	OLA 9	---	SNVT_volt_f	---	nvoAnalogIn_8
●	NVI	OLA 10	---	SNVT_volt_f	---	nvoAnalogIn_9
●	NVI	OLA 11	---	SNVT_volt_f	---	nvoAnalogIn_10
●	NVI	OLA 12				11
●	NVI	OLA 13				12
●	NVI	OLA 14				13
●	NVI	OLA 15				14
●	NVI	OLA 16				15
●	NVI	OLA 17				16
●	NVI	OLA 18				1
●	NVI	OLA 19				2
●	NVO	OLS 1				1
●	NVO	OLS 2				2
●	NVO	OLS 3	---	SNVT_count_f	---	nviAnalogOut_3
●	NVO	OLS 4	---	SNVT_count_f	---	nviAnalogOut_4
●	NVO	OLS 5	---	SNVT_switch	---	nviDiscreteOut_1
●	NVO	OLS 5		u value	nvt_cat_unsigned_short	nviDiscreteOut_1_1

BB2-LON Configuration

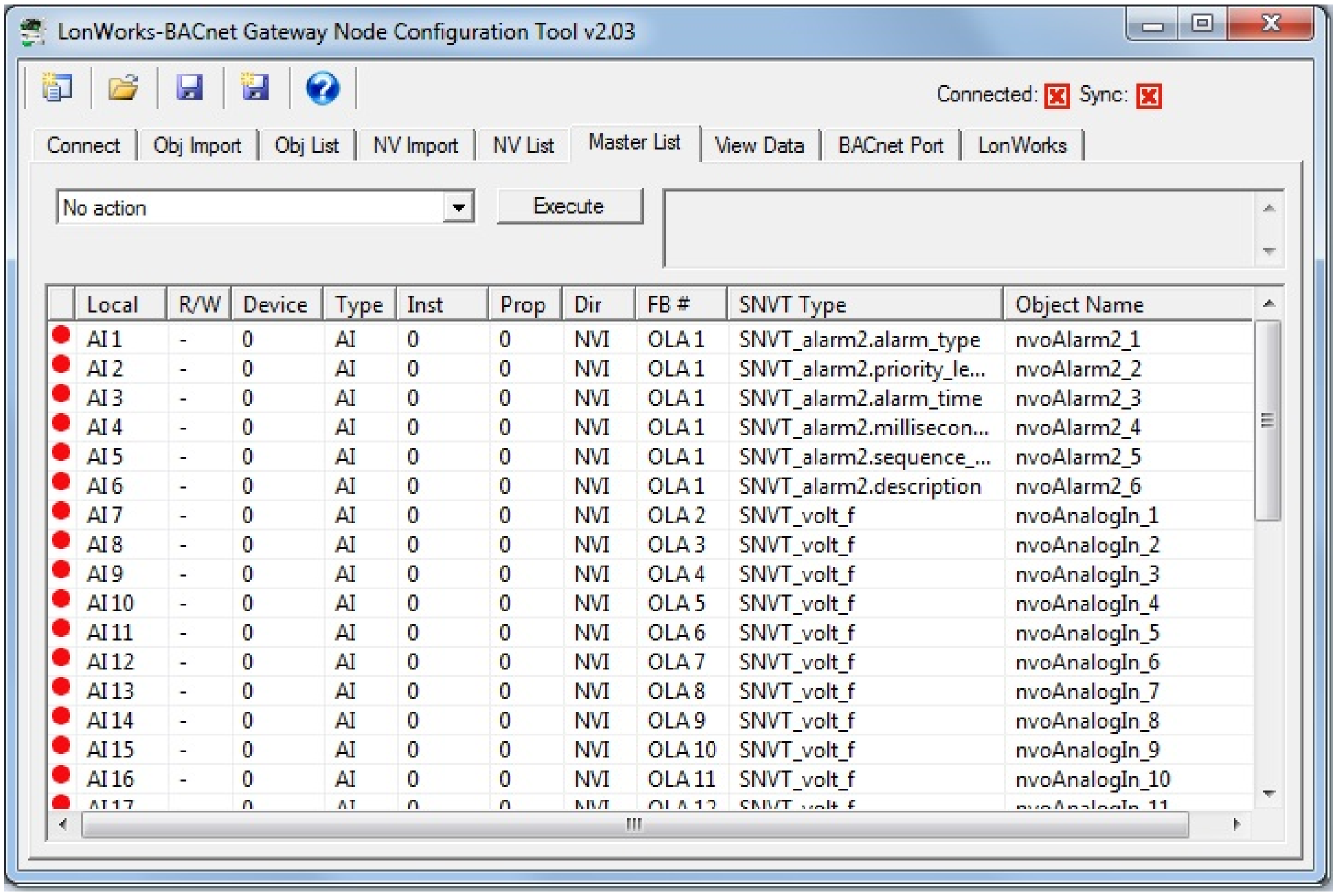
Convert instances of SNVT_switch to single BACnet object?
Click OK to convert to one.

After data object assignment, the Loc... (Local Objects) column will be populated.



	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVI	OLA 9	AI 14	SNVT_volt_f	---	nvoAnalogIn_8
●	NVI	OLA 10	AI 15	SNVT_volt_f	---	nvoAnalogIn_9
●	NVI	OLA 11	AI 16	SNVT_volt_f	---	nvoAnalogIn_10
●	NVI	OLA 12	AI 17	SNVT_volt_f	---	nvoAnalogIn_11
●	NVI	OLA 13	AI 18	SNVT_volt_f	---	nvoAnalogIn_12
●	NVI	OLA 14	AI 19	SNVT_volt_f	---	nvoAnalogIn_13
●	NVI	OLA 15	AI 20	SNVT_volt_f	---	nvoAnalogIn_14
●	NVI	OLA 16	AI 21	SNVT_volt_f	---	nvoAnalogIn_15
●	NVI	OLA 17	AI 22	SNVT_volt_f	---	nvoAnalogIn_16
●	NVI	OLA 18	AI 23	SNVT_count_f	---	nvoDiscreteIn_1
●	NVI	OLA 19	AI 24	SNVT_count_f	---	nvoDiscreteIn_2
●	NVO	OLS 1	AO 1	SNVT_count_f	---	nviAnalogOut_1
●	NVO	OLS 2	AO 2	SNVT_count_f	---	nviAnalogOut_2
●	NVO	OLS 3	AO 3	SNVT_count_f	---	nviAnalogOut_3
●	NVO	OLS 4	AO 4	SNVT_count_f	---	nviAnalogOut_4
●	NVO	OLS 5	---	SNVT_switch	---	nviDiscreteOut_1
●	NVO	OLS 5	AO 5	.value	nvt_cat_unsigned_short	nviDiscreteOut_1

The results of the configuration process are now available on the Master List page. From here, you would proceed to send NV definitions to the device, followed by data object definitions, as illustrated toward the end of the example given above. Once the Master List is complete, the process of sending configuration to the gateway device is the same regardless of approach used to build it.



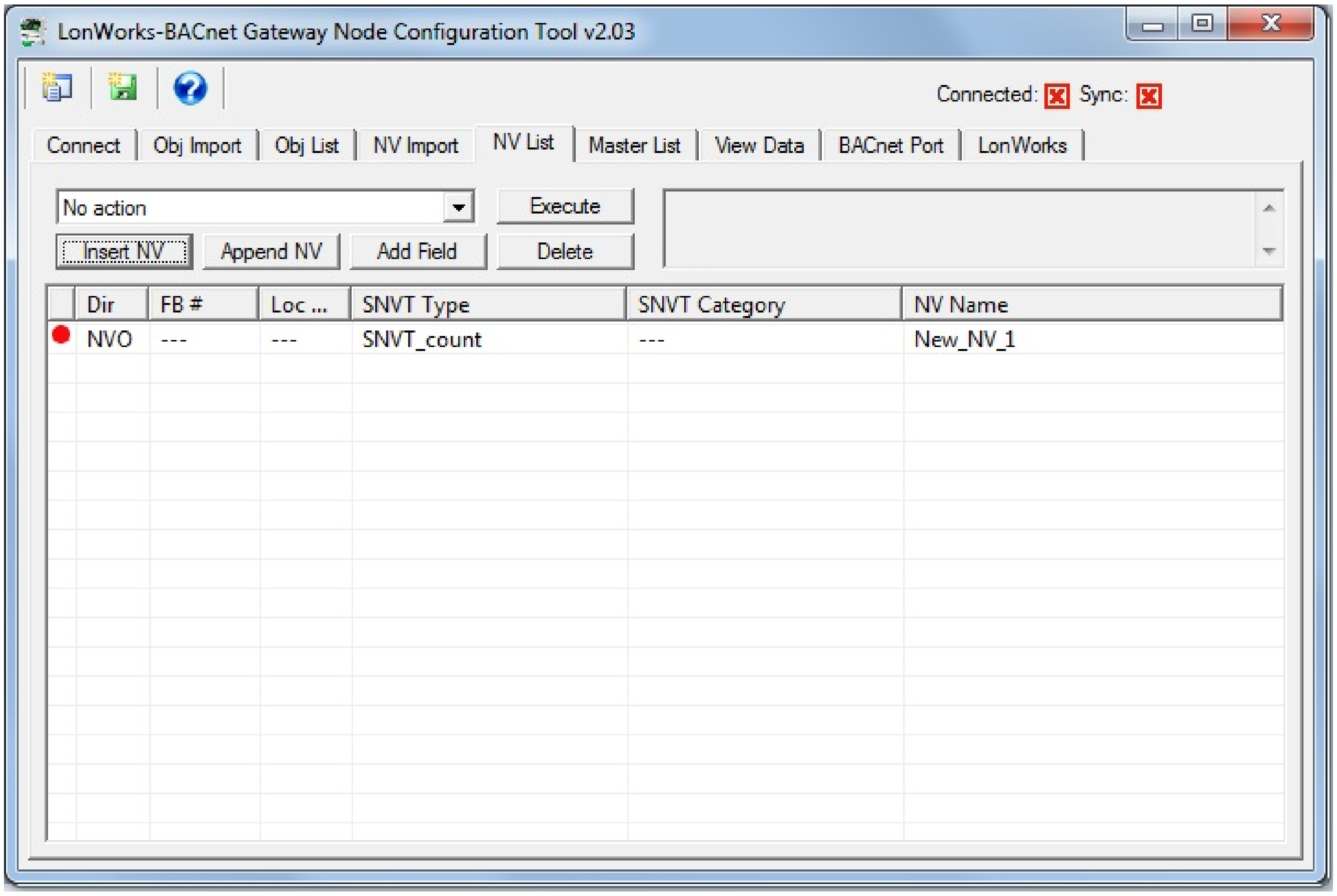
	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	SNVT Type	Object Name
●	AI 1	-	0	AI	0	0	NVI	OLA 1	SNVT_alarm2.alarm_type	nvoAlarm2_1
●	AI 2	-	0	AI	0	0	NVI	OLA 1	SNVT_alarm2.priority_le...	nvoAlarm2_2
●	AI 3	-	0	AI	0	0	NVI	OLA 1	SNVT_alarm2.alarm_time	nvoAlarm2_3
●	AI 4	-	0	AI	0	0	NVI	OLA 1	SNVT_alarm2.millisecon...	nvoAlarm2_4
●	AI 5	-	0	AI	0	0	NVI	OLA 1	SNVT_alarm2.sequence_...	nvoAlarm2_5
●	AI 6	-	0	AI	0	0	NVI	OLA 1	SNVT_alarm2.description	nvoAlarm2_6
●	AI 7	-	0	AI	0	0	NVI	OLA 2	SNVT_volt_f	nvoAnalogIn_1
●	AI 8	-	0	AI	0	0	NVI	OLA 3	SNVT_volt_f	nvoAnalogIn_2
●	AI 9	-	0	AI	0	0	NVI	OLA 4	SNVT_volt_f	nvoAnalogIn_3
●	AI 10	-	0	AI	0	0	NVI	OLA 5	SNVT_volt_f	nvoAnalogIn_4
●	AI 11	-	0	AI	0	0	NVI	OLA 6	SNVT_volt_f	nvoAnalogIn_5
●	AI 12	-	0	AI	0	0	NVI	OLA 7	SNVT_volt_f	nvoAnalogIn_6
●	AI 13	-	0	AI	0	0	NVI	OLA 8	SNVT_volt_f	nvoAnalogIn_7
●	AI 14	-	0	AI	0	0	NVI	OLA 9	SNVT_volt_f	nvoAnalogIn_8
●	AI 15	-	0	AI	0	0	NVI	OLA 10	SNVT_volt_f	nvoAnalogIn_9
●	AI 16	-	0	AI	0	0	NVI	OLA 11	SNVT_volt_f	nvoAnalogIn_10
●	AI 17	-	0	AI	0	0	NVI	OLA 12	SNVT_volt_f	nvoAnalogIn_11

3.4 Build Configuration from Scratch

It is not required that you start with either a CSV file or an XIF file. You can use the configuration tool to start with a blank slate and build your configuration from scratch.

It is assumed that you have some familiarity with BACnet, and also an understanding of LonWorks network variables. You will need to obtain a copy of the documentation of SNVT types from LonMark (www.lonmark.org) in order to have any success in creating the LonWorks side of the gateway configuration.

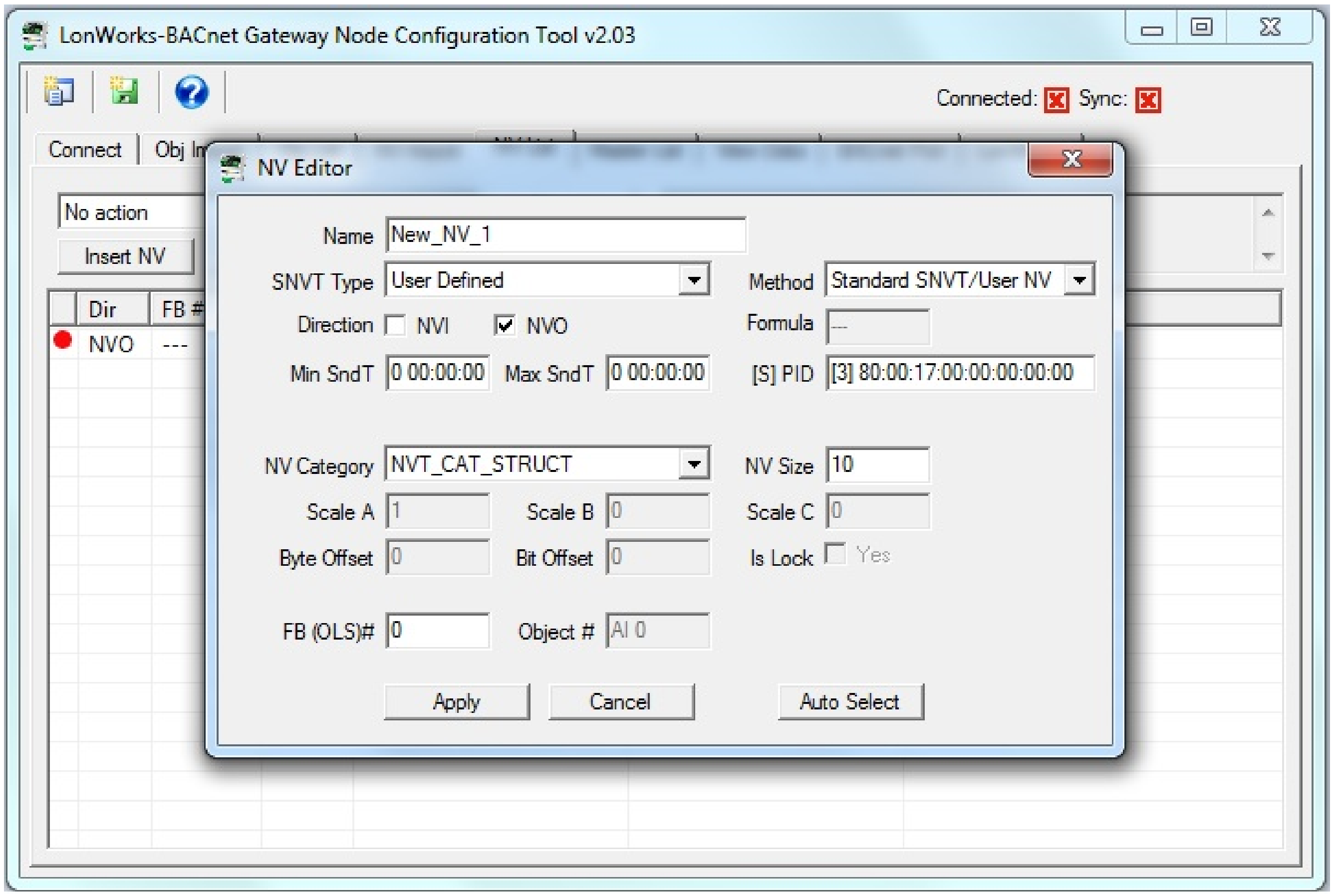
For each network variable you wish to add, click Insert NV.



	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVO	---	---	SNVT_count	---	New_NV_1

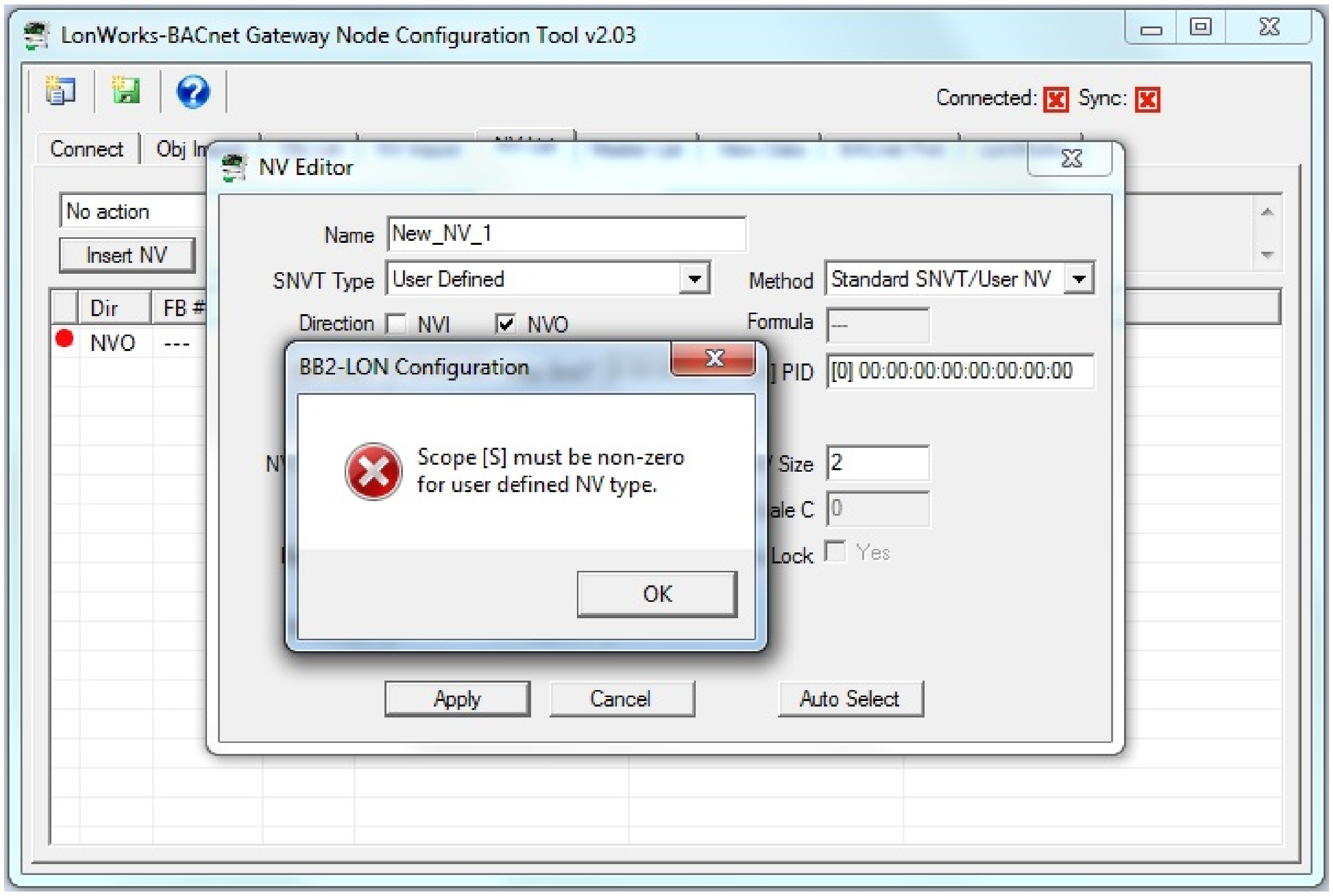
The newly inserted NV will default to SNVT_count. If you wish to change it, double click on the line to be changed, and the NV Editor dialog will appear. In the example that follows, we will complicate things to the maximum by creating a user defined structured network variable.

When creating a structure, the NV needs to be set to NV Category NVT_CAT_STRUCT and the NV Size needs to be set to the number of bytes that make up this variable in the LonWorks device. You also need to change the Scope [S] and PID (program ID) to something unique that defines the manufacturer of the device you are conforming to.



Name: New_NV_1
SNVT Type: User Defined
Method: Standard SNVT/User NV
Direction: NVI [✓ NVO]
Formula: ---
Min SndT: 0 00:00:00
Max SndT: 0 00:00:00
[S] PID: [3] 80:00:17:00:00:00:00:00
NV Category: NVT_CAT_STRUCT
NV Size: 10
Scale A: 1
Scale B: 0
Scale C: 0
Byte Offset: 0
Bit Offset: 0
Is Lock: Yes
FB (OLS)#: 0
Object #: AI 0

If you do not change the scope, most commonly to 3 for 'manufacturer defined', you will get an error message and not be allowed to proceed. If the scope remains at zero, then the gateway (and all other LonWorks devices on the network) will attempt to interpret your NV as a standard LonMark type and data conversion results will be wrong.

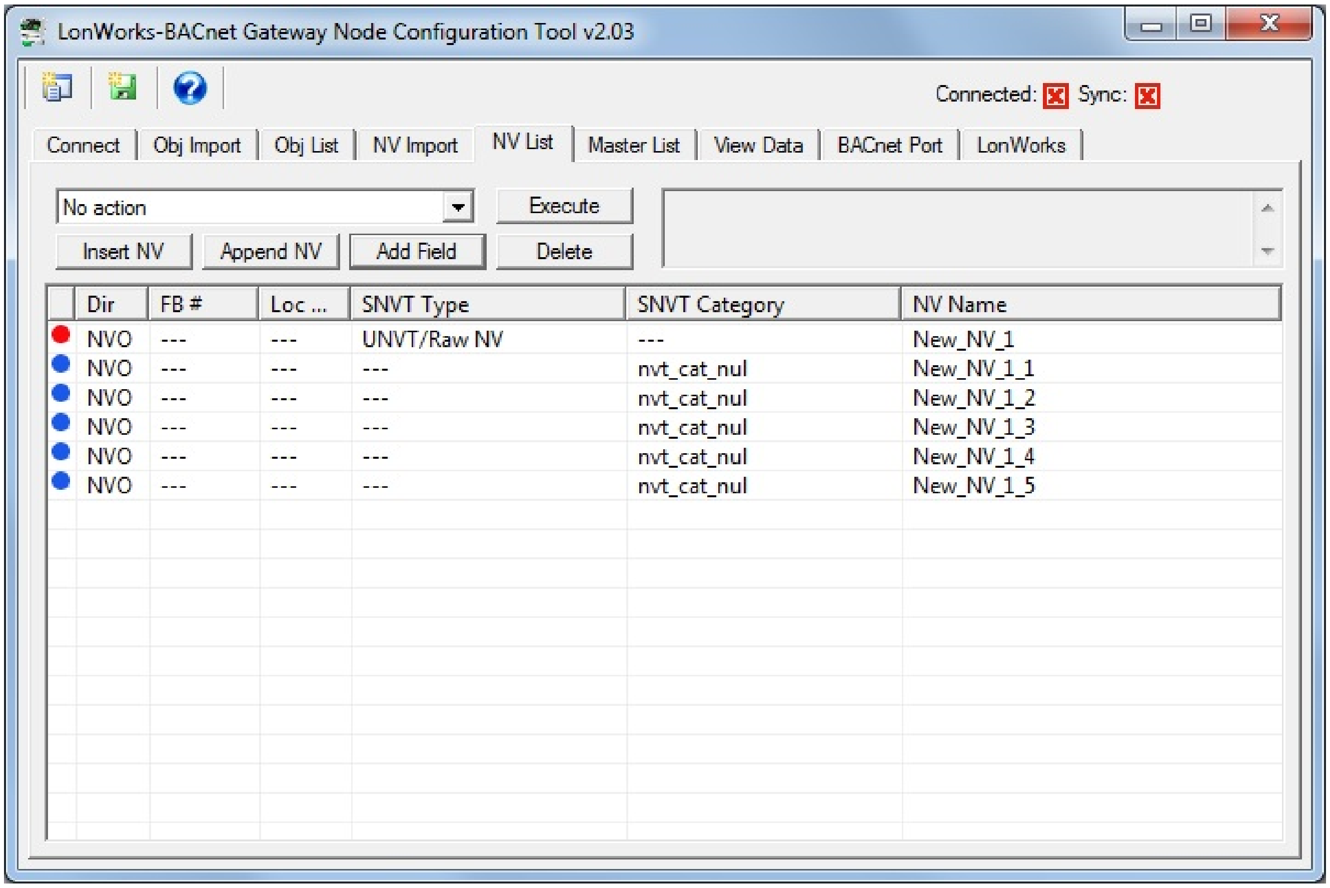


Name New_NV_1
SNVT Type User Defined
Method Standard SNVT/User NV
Direction ☐ NVI ☑ NVO
Formula ---
PID [0] 00:00:00:00:00:00:00:00
Size 2
Lock ☐ Yes

BB2-LON Configuration

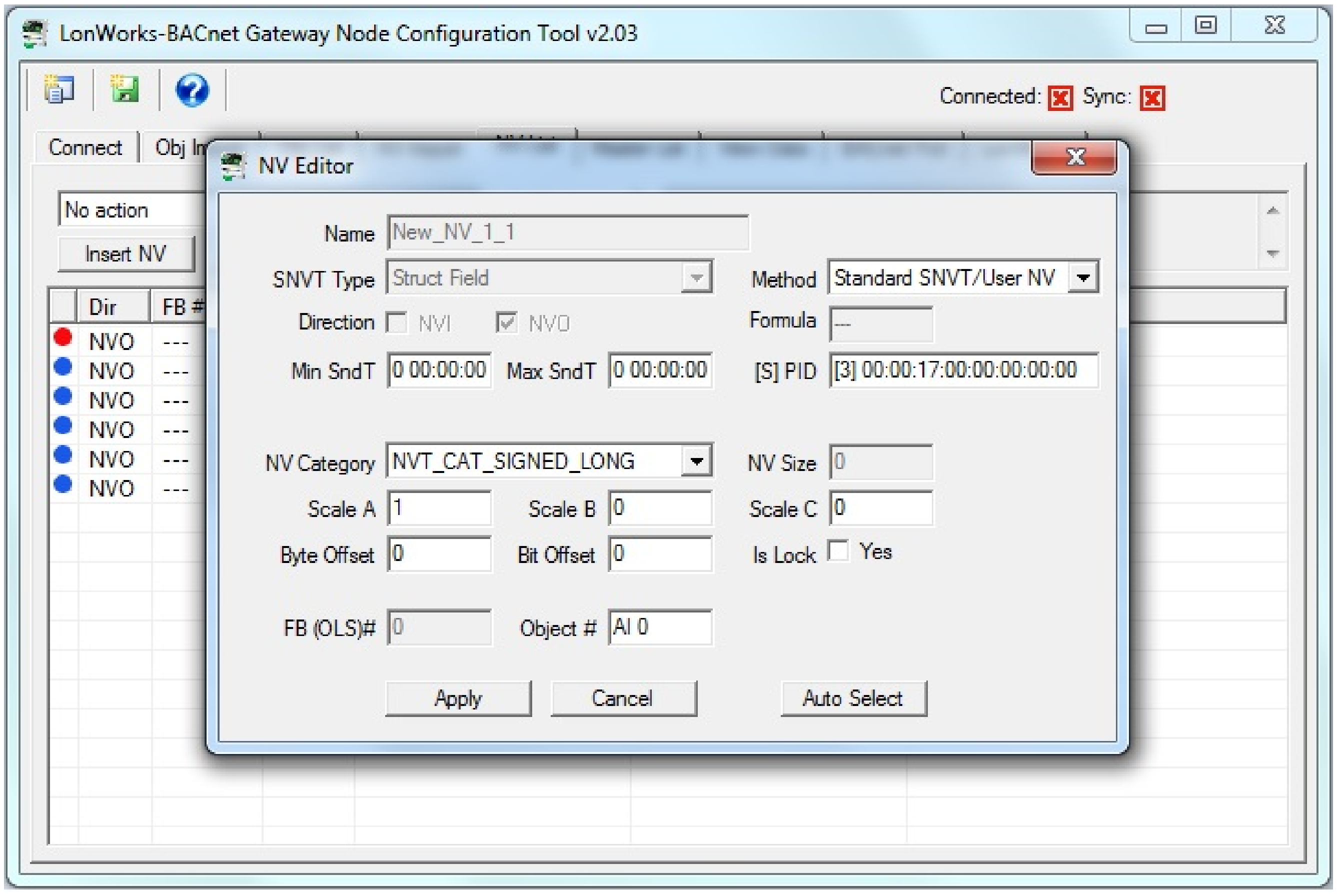
Scope [S] must be non-zero for user defined NV type.

After defining the NV as a structure, you next need to add fields to the structure. Do this by clicking Add Field.



	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVO	---	---	UNVT/Raw NV	---	New_NV_1
●	NVO	---	---	---	nvt_cat_nul	New_NV_1_1
●	NVO	---	---	---	nvt_cat_nul	New_NV_1_2
●	NVO	---	---	---	nvt_cat_nul	New_NV_1_3
●	NVO	---	---	---	nvt_cat_nul	New_NV_1_4
●	NVO	---	---	---	nvt_cat_nul	New_NV_1_5

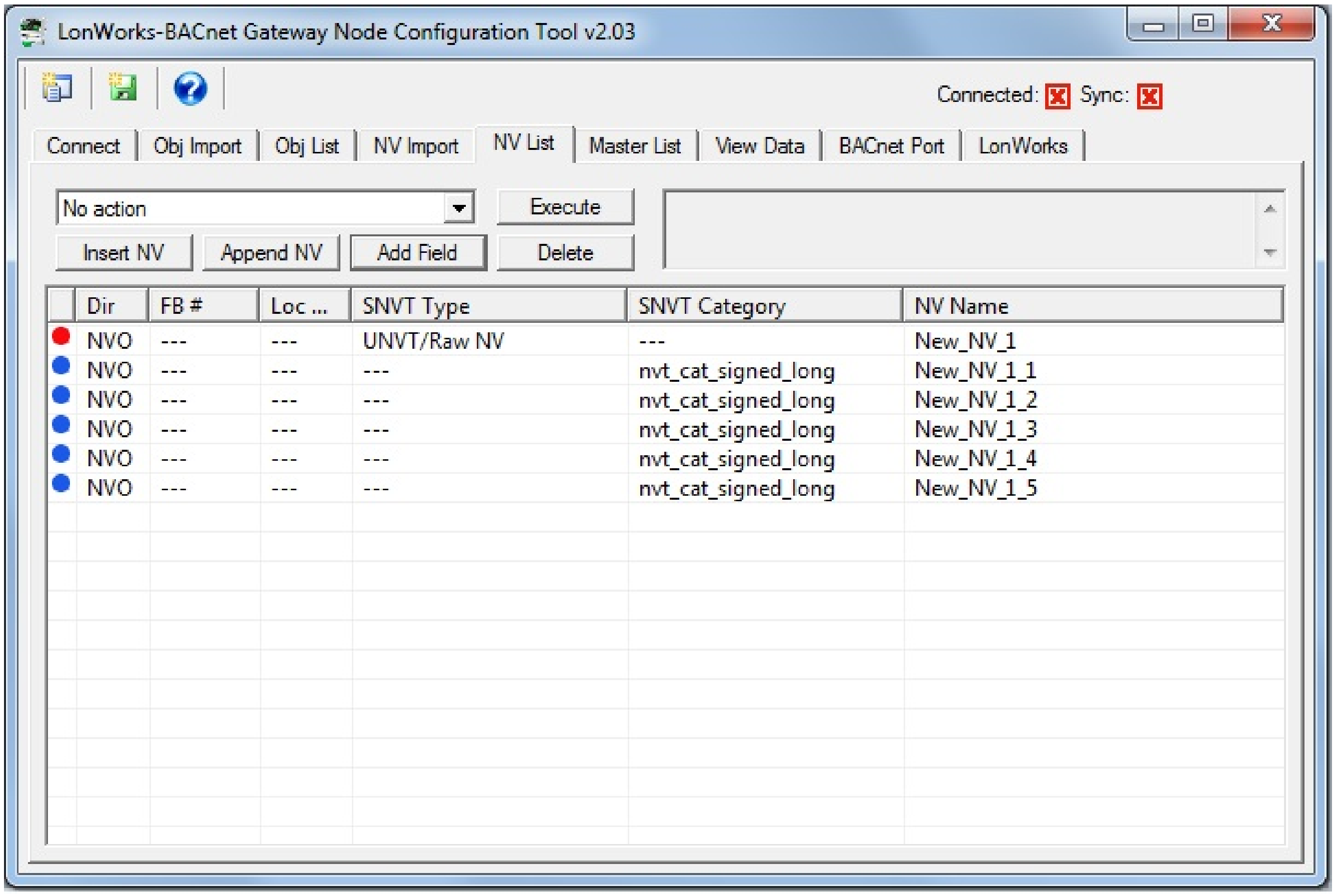
Next, edit each field by double clicking on that line. Specify the data type for the field. It is also important to specify the Byte Offset in the structure. An offset of zero means this field occupies the first byte(s) of the structure. You also need to enter the scale values. These follow the LonMark definition of scale.



Dir	FB #
● NVO	---
● NVO	---
● NVO	---
● NVO	---
● NVO	---
● NVO	---

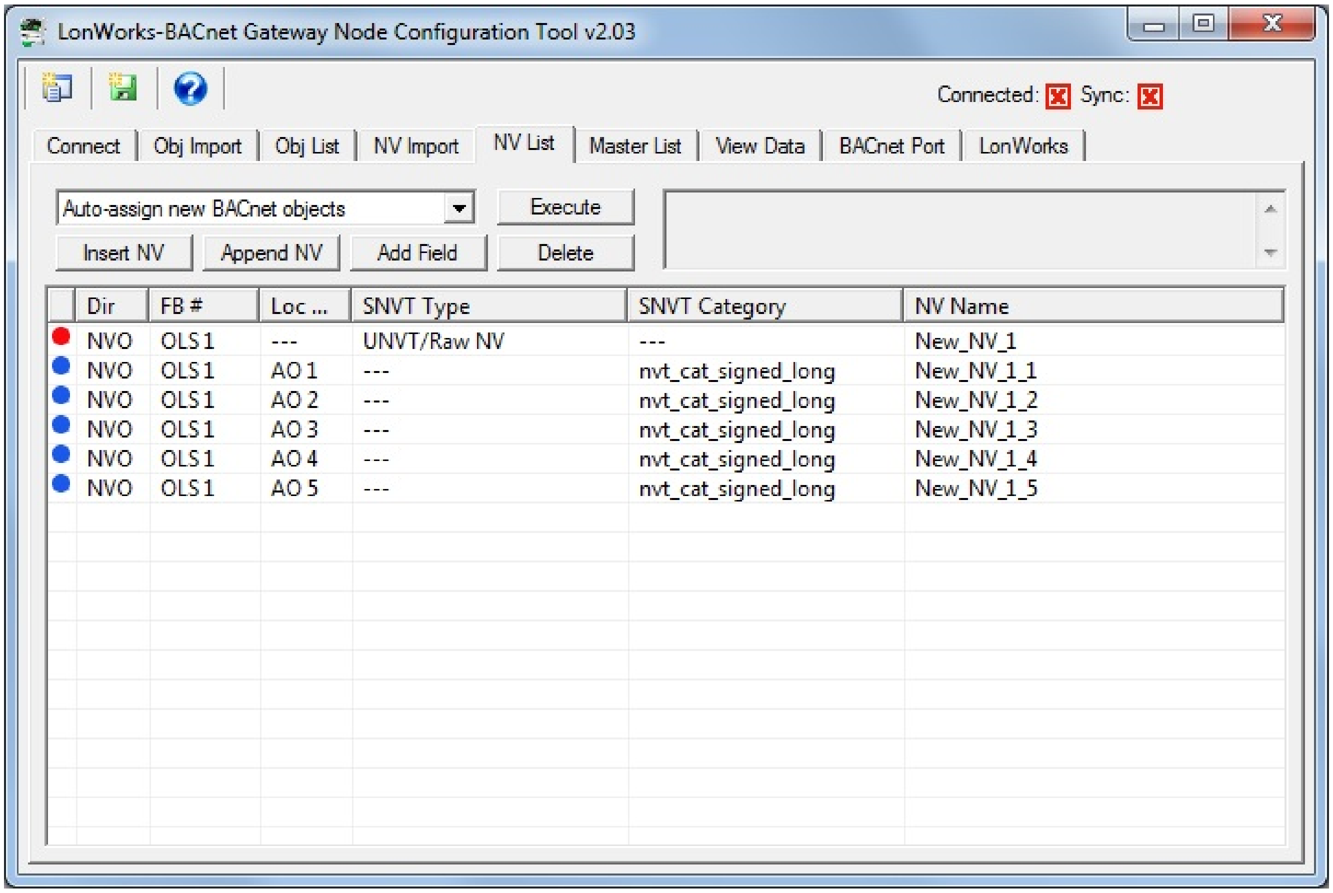
Name: New_NV_1_1
SNVT Type: Struct Field
Method: Standard SNVT/User NV
Direction: ☐ NVI ☑ NVO
Formula: ---
Min SndT: 0 00:00:00
Max SndT: 0 00:00:00
[S] PID: [3] 00:00:17:00:00:00:00:00
NV Category: NVT_CAT_SIGNED_LONG
NV Size: 0
Scale A: 1
Scale B: 0
Scale C: 0
Byte Offset: 0
Bit Offset: 0
Is Lock: ☐ Yes
FB (OLS)#: 0
Object #: AI 0

Following configuration of the NV and its fields, your NV List will appear as follows for this first single NV that will map to five BACnet objects.



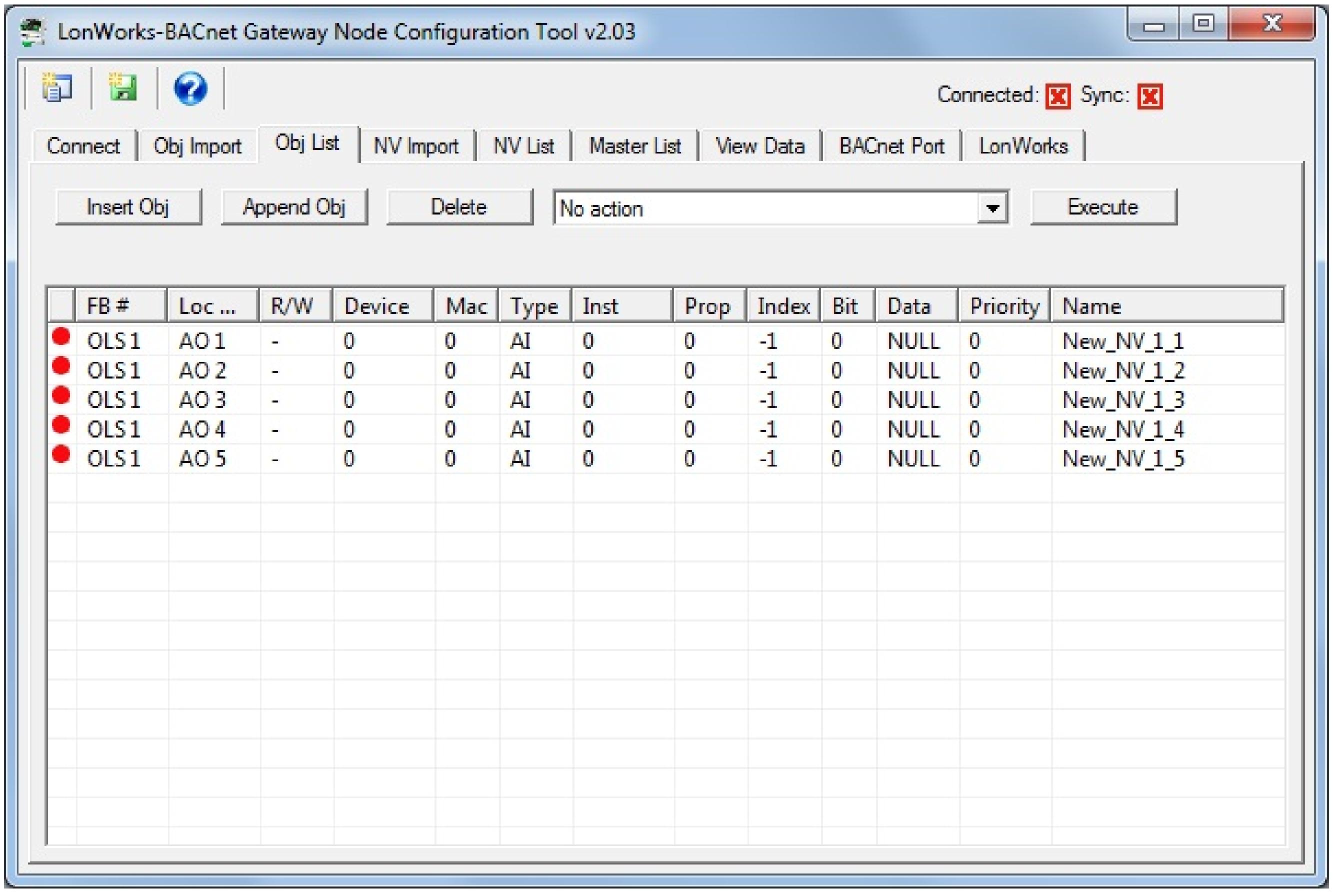
	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVO	---	---	UNVT/Raw NV	---	New_NV_1
●	NVO	---	---	---	nvt_cat_signed_long	New_NV_1_1
●	NVO	---	---	---	nvt_cat_signed_long	New_NV_1_2
●	NVO	---	---	---	nvt_cat_signed_long	New_NV_1_3
●	NVO	---	---	---	nvt_cat_signed_long	New_NV_1_4
●	NVO	---	---	---	nvt_cat_signed_long	New_NV_1_5

If you have no further network variables to add, proceed with the auto-build of the rest of the configuration. First execute "Auto-assign new NV function blocks". Then execute "Auto-assign new BACnet objects". Upon completion of these steps, the NV List would appear as follows for this example.



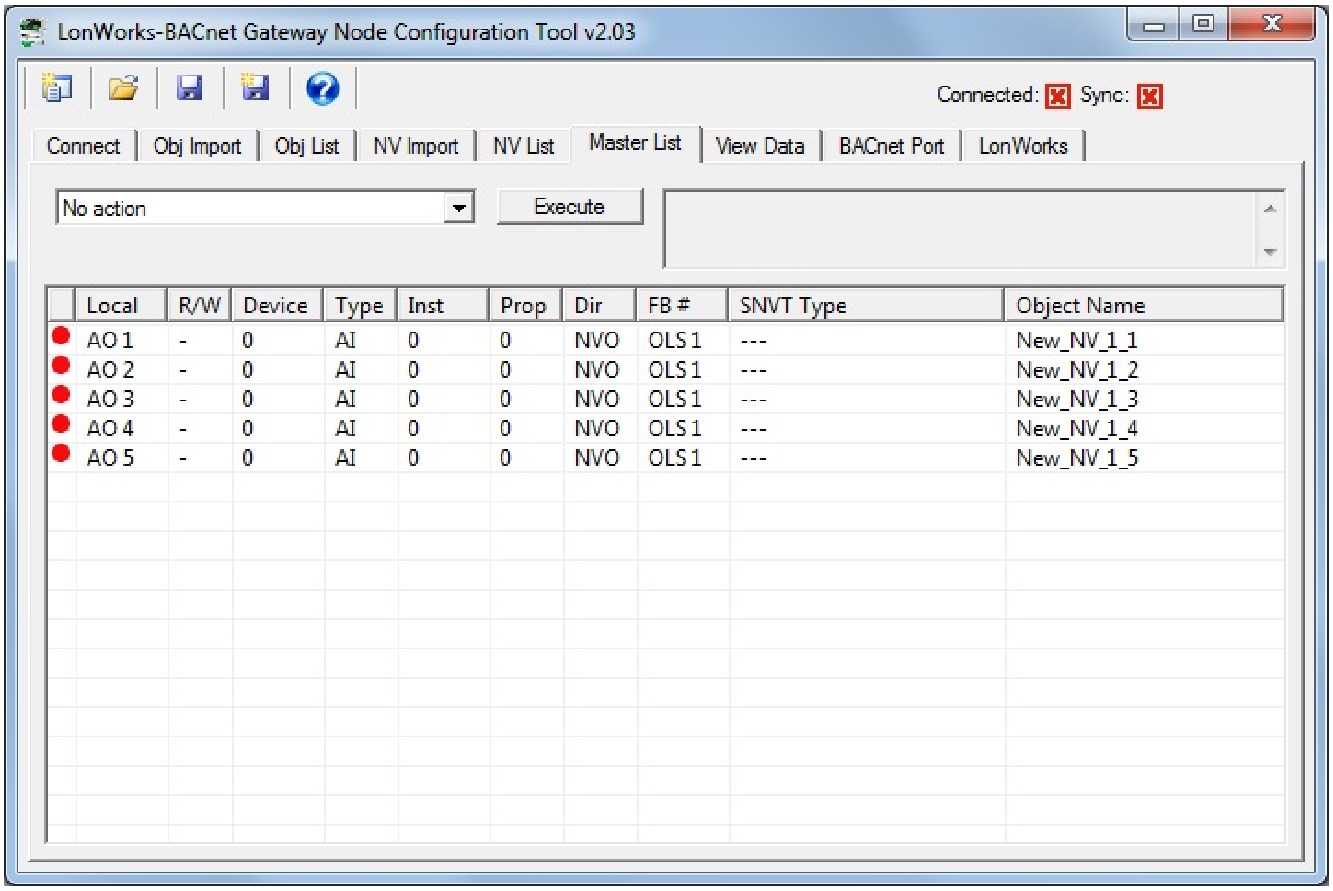
	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVO	OLS 1	---	UNVT/Raw NV	---	New_NV_1
●	NVO	OLS 1	AO 1	---	nvt_cat_signed_long	New_NV_1_1
●	NVO	OLS 1	AO 2	---	nvt_cat_signed_long	New_NV_1_2
●	NVO	OLS 1	AO 3	---	nvt_cat_signed_long	New_NV_1_3
●	NVO	OLS 1	AO 4	---	nvt_cat_signed_long	New_NV_1_4
●	NVO	OLS 1	AO 5	---	nvt_cat_signed_long	New_NV_1_5

The corresponding Obj List (BACnet object list) will appear as follows for this example.



	FB #	Loc ...	R/W	Device	Mac	Type	Inst	Prop	Index	Bit	Data	Priority	Name
●	OLS 1	AO 1	-	0	0	AI	0	0	-1	0	NULL	0	New_NV_1_1
●	OLS 1	AO 2	-	0	0	AI	0	0	-1	0	NULL	0	New_NV_1_2
●	OLS 1	AO 3	-	0	0	AI	0	0	-1	0	NULL	0	New_NV_1_3
●	OLS 1	AO 4	-	0	0	AI	0	0	-1	0	NULL	0	New_NV_1_4
●	OLS 1	AO 5	-	0	0	AI	0	0	-1	0	NULL	0	New_NV_1_5

The Master List will appear as follows for this example. At this point, you would execute the "Send NV definitions to device" followed by "Send Object maps to device" as in the previous examples.



	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	SNVT Type	Object Name
●	AO 1	-	0	AI	0	0	NVO	OLS 1	---	New_NV_1_1
●	AO 2	-	0	AI	0	0	NVO	OLS 1	---	New_NV_1_2
●	AO 3	-	0	AI	0	0	NVO	OLS 1	---	New_NV_1_3
●	AO 4	-	0	AI	0	0	NVO	OLS 1	---	New_NV_1_4
●	AO 5	-	0	AI	0	0	NVO	OLS 1	---	New_NV_1_5

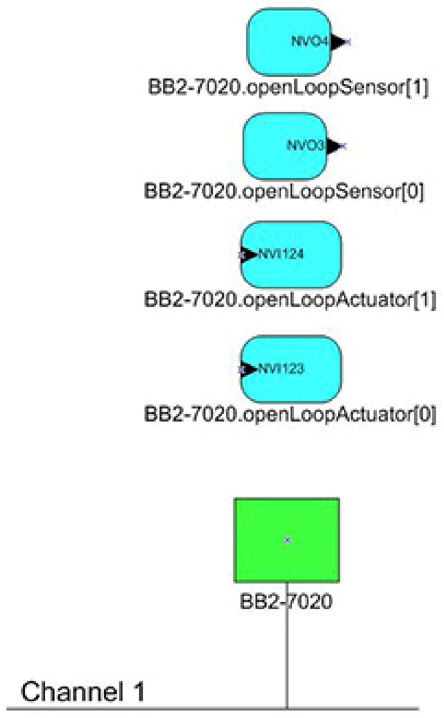
3.5 Commissioning Gateway as Node on Network

Even without any configuration, the Babel Buster LonWorks gateway will appear as a valid LonWorks node on the network. Without any configuration, all of the network variables default to a UNVTrawData type of variable. The only purpose of this variable type is to reserve the necessary memory space to provide storage for the largest NV type possible (31 bytes). Network variables that still have this default UNVT type will be regarded as 'unused' by the configuration tool. But they will still appear and be available to any network management tool on the LonWorks side.

Configuration will generally include changing the NV types to something more meaningful to your application. You will generally want to do as much pre-configuring of the gateway as possible before commissioning it on the LonWorks network. When you are ready to commission, follow the same procedure you would for commissioning any other LonWorks node using your favorite network management tool.

If you are commissioning the gateway using LonMaker, start by placing a device on the drawing and commissioning it following the standard practices. You can import the XIF from the device as well as import configuration properties from the device.

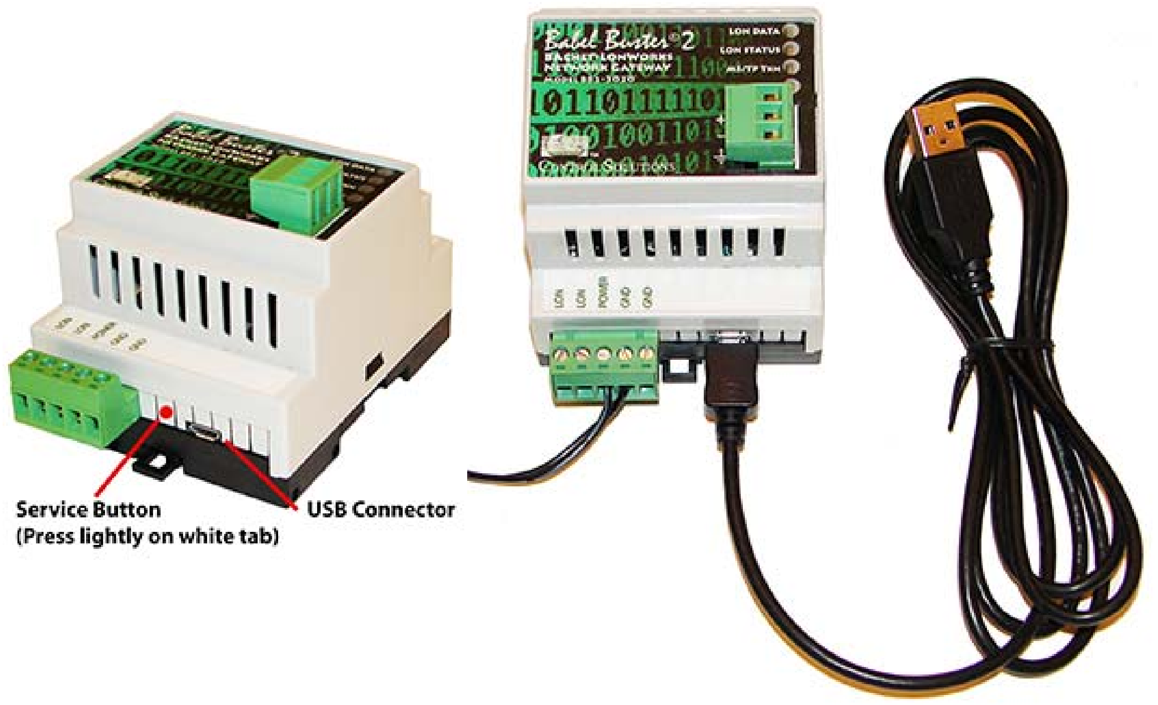
Once the device is on your drawing, start placing function blocks as needed to expose the various network variables. Then draw connectors to bind network variables as applicable.


NVO4
BB2-7020.openLoopSensor[1]
NVO3
BB2-7020.openLoopSensor[0]
NVI124
BB2-7020.openLoopActuator[1]
NVI123
BB2-7020.openLoopActuator[0]
BB2-7020
Channel 1

4 Tool 'Connect' Page

4.1 Connecting Configuration Tool to Gateway Device

The Babel Buster 2 LonWorks Gateway includes a USB port for configuration of the gateway. This eliminates any conflicts with the communications on either the BACnet or LonWorks ports and does not even require either network to be functional in order to configure the gateway. The only interface required is a USB cable. You do need to install the USB driver the first time you connect a Babel Buster via USB. After that, the gateway simply shows up as a COM port since USB is simulating a serial port for purposes of configuring the gateway.



Use your PC's device manager (found in the control panel) to locate the COM port that the gateway appears on once connected via USB. Select this COM port on the Connect page of the tool, and click Connect.

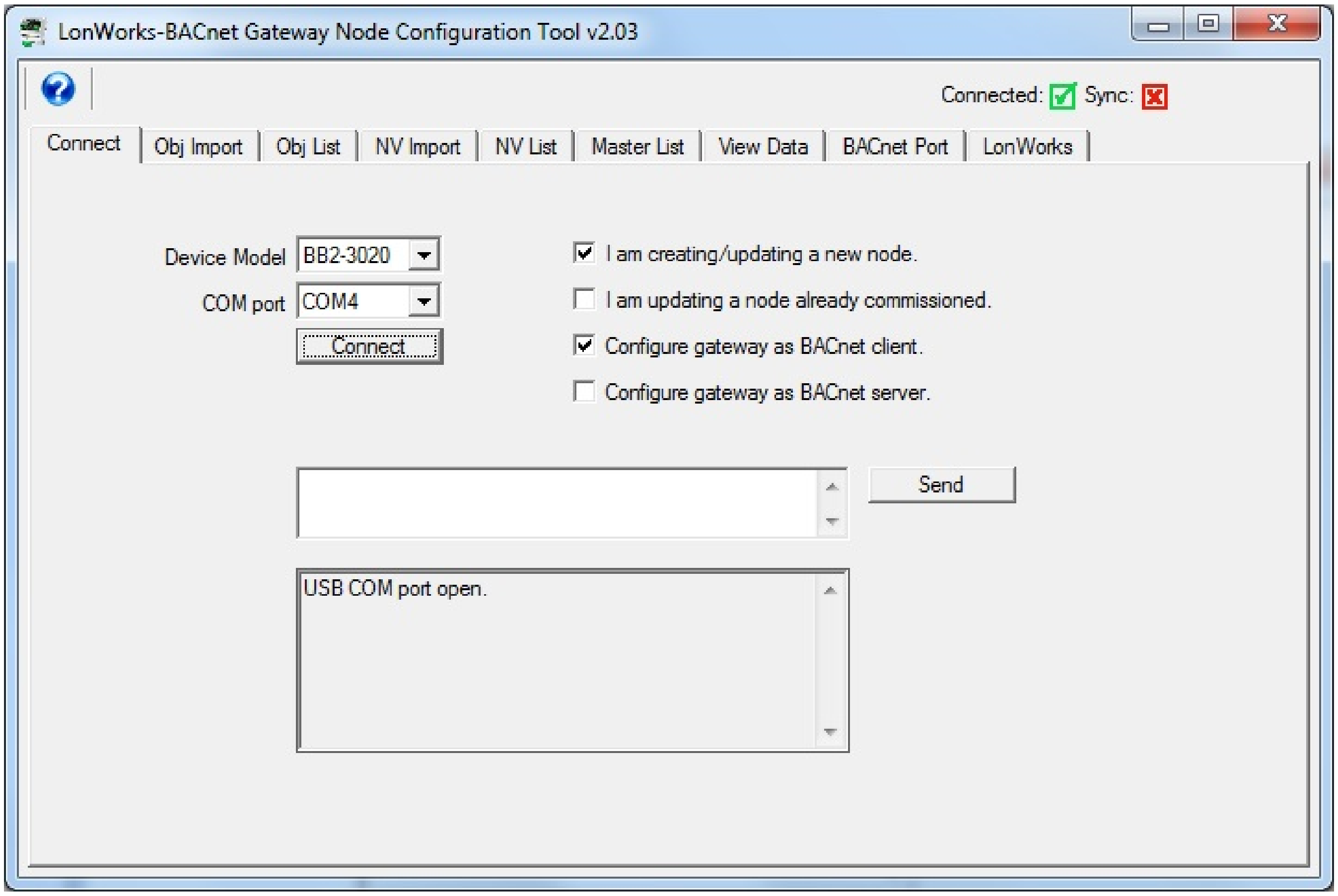
Most of the configuration will be identical for BB2-3020 and BB2-7020. The only significant difference is that the IP settings will show up instead of MS/TP settings on the BACnet Port page when BB2-6020 is selected as Device Model.

If you are configuring a new gateway for the first time, select the check box indicating that. If you are updating a gateway already commissioned (installed on LonWorks network), then check that box. Doing so will prevent you from making changes that will cause your LNS database to become out of sync with the device.

The configuration will default to BACnet client. Check the 'Configure as BACnet server' box if you will be setting up the Babel Buster gateway as being the BACnet server device, expecting an external BACnet client to query it. Some of the object related configuration changes based on whether the gateway will be BACnet client or server.

IMPORTANT: If you have connected USB to the gateway, and then power cycle the gateway, the USB connection will be lost. You will need to unplug the USB cable from the gateway and reconnect, then reconnect the configuration tool. It may also sometimes be necessary to close and restart the configuration tool software to re-establish a lost USB connection.

IMPORTANT: If you select a COM port and 'connect' but do not get any response in the configuration tool, it is possible you opened a COM port that is a valid port but is not the USB port for the gateway. When this happens, the tool will get hung up waiting for a response. At the same time, Windows gets confused and doesn't let the configuration tool time out either. If you are stuck in this state, close the configuration tool, forcefully via the system task manager if necessary, and restart - including disconnecting and reconnecting the USB cable.


LonWorks-BACnet Gateway Node Configuration Tool v2.03

Connected: ✓ Sync: ✗

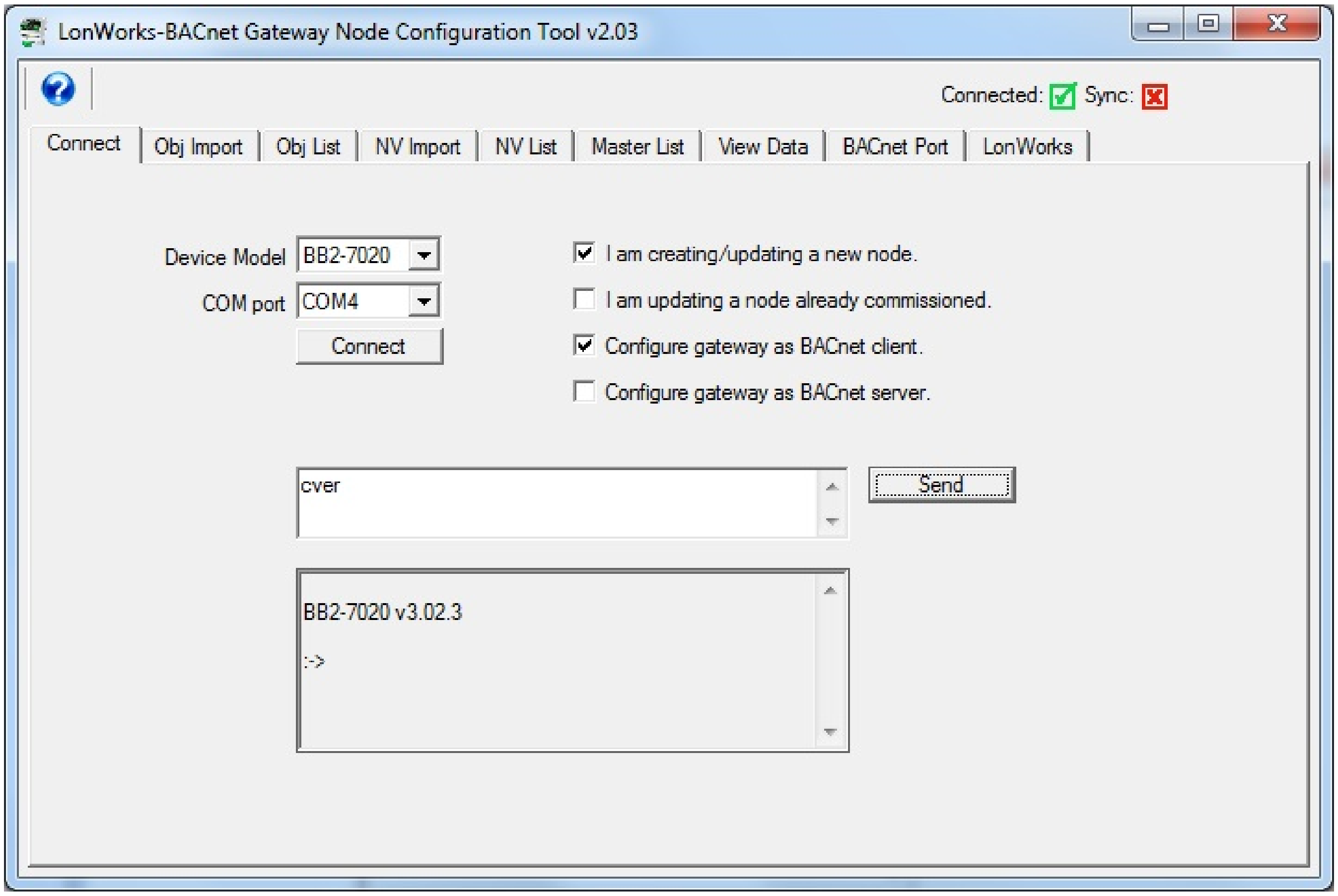
Device Model: BB2-3020
COM port: COM4

[✓] I am creating/updating a new node.
[ ] I am updating a node already commissioned.
[✓] Configure gateway as BACnet client.
[ ] Configure gateway as BACnet server.

USB COM port open.

The "Connected" box will turn green if the COM port is simply able to be opened. It does not actually mean communications are successful. To test that, type the command "cver" in the command window, and click Send. This will send a request for firmware version to the device. If successful, you will see something comparable to the example below.

The command window is also used for a number of other diagnostic commands. Refer to the section on "Diagnostics via the USB Console" for more details. Normally, these diagnostic commands will not be required, but can be helpful in some instances of trouble shooting.


LonWorks-BACnet Gateway Node Configuration Tool v2.03

Connected: ✓ Sync: ✗

Device Model BB2-7020

COM port COM4

[✓] I am creating/updating a new node.

[ ] I am updating a node already commissioned.

[✓] Configure gateway as BACnet client.

[ ] Configure gateway as BACnet server.

```
cver
```

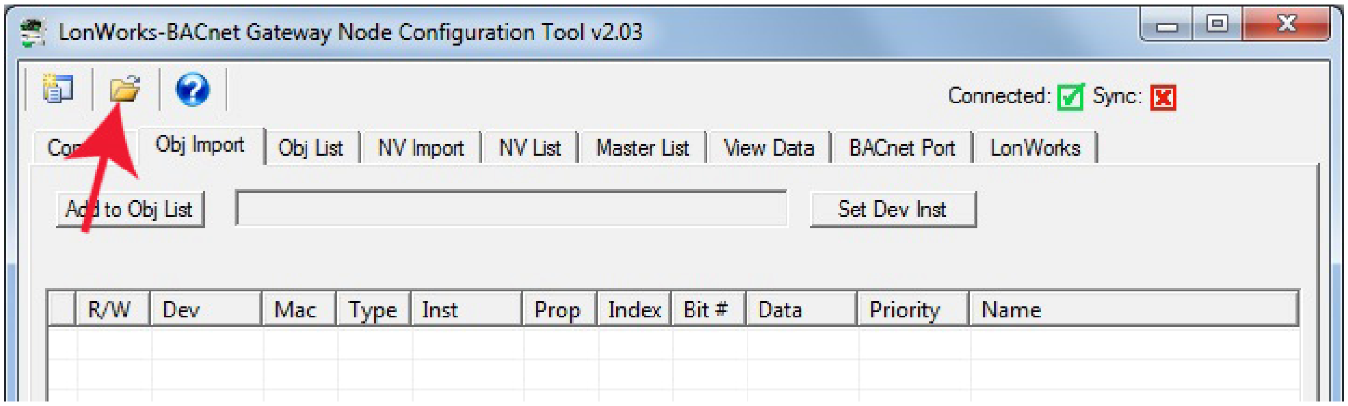
```
BB2-7020 v3.02.3

:->
```

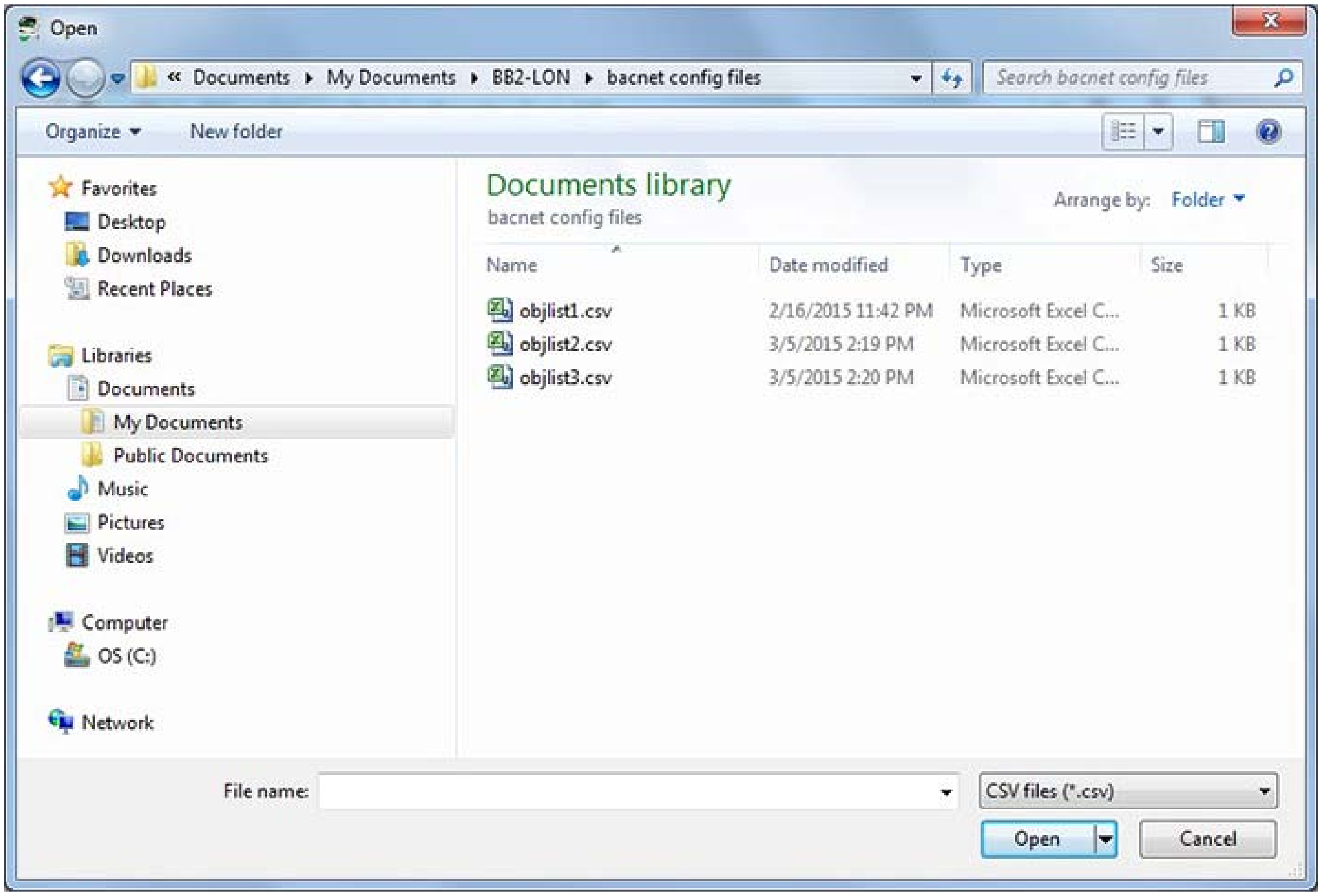
5 Tool 'Obj Import' Page

5.1 Importing a CSV Object List

To import a BACnet object list from a CSV file, start by going to the Obj Import page, then click on the file open icon.



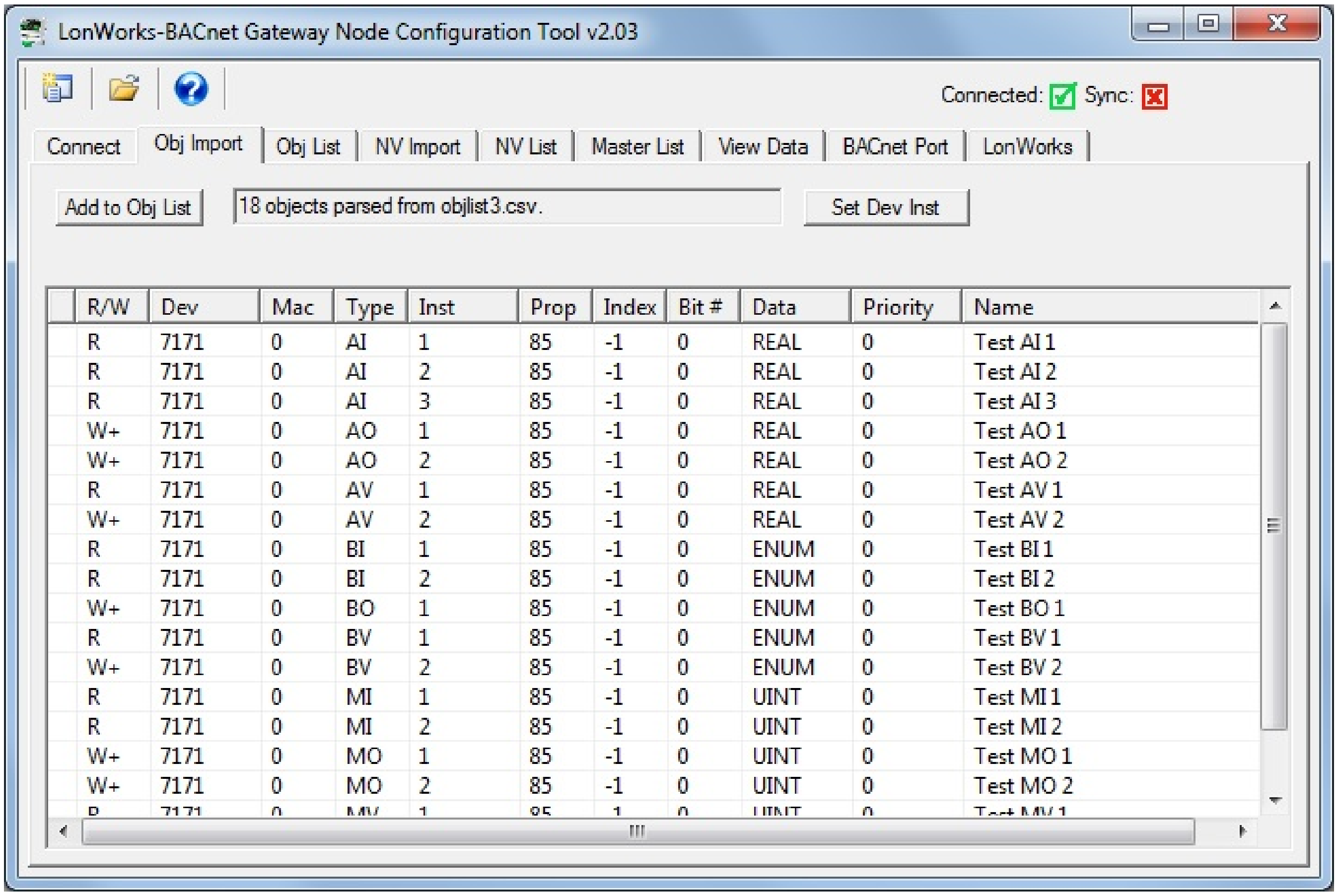
You will see the familiar file open dialog. Select your CSV file.



Name	Date modified	Type	Size
objlist1.csv	2/16/2015 11:42 PM	Microsoft Excel C...	1 KB
objlist2.csv	3/5/2015 2:19 PM	Microsoft Excel C...	1 KB
objlist3.csv	3/5/2015 2:20 PM	Microsoft Excel C...	1 KB

The format required for the CSV file is described in Appendix D of this user guide. For reference, the content of the objlist3.csv file loaded here contains the following CSV list, and upon loading, will appear as shown in the next screen shot.

```
RW,DEV,OBJTYPE,OBJINST,PROP,DATA,NAME
R,7171,AI,1,85,REAL,Test AI 1
R,7171,AI,2,85,REAL,Test AI 2
R,7171,AI,3,85,REAL,Test AI 3
W+,7171,AO,1,85,REAL,Test AO 1
W+,7171,AO,2,85,REAL,Test AO 2
R,7171,AV,1,85,REAL,Test AV 1
W+,7171,AV,2,85,REAL,Test AV 2
R,7171,BI,1,85,ENUM,Test BI 1
R,7171,BI,2,85,ENUM,Test BI 2
W+,7171,BO,1,85,ENUM,Test BO 1
R,7171,BV,1,85,ENUM,Test BV 1
W+,7171,BV,2,85,ENUM,Test BV 2
R,7171,MI,1,85,UINT,Test MI 1
R,7171,MI,2,85,UINT,Test MI 2
W+,7171,MO,1,85,UINT,Test MO 1
W+,7171,MO,2,85,UINT,Test MO 2
R,7171,MV,1,85,UINT,Test MV 1
W+,7171,MV,2,85,UINT,Test MV 2
```



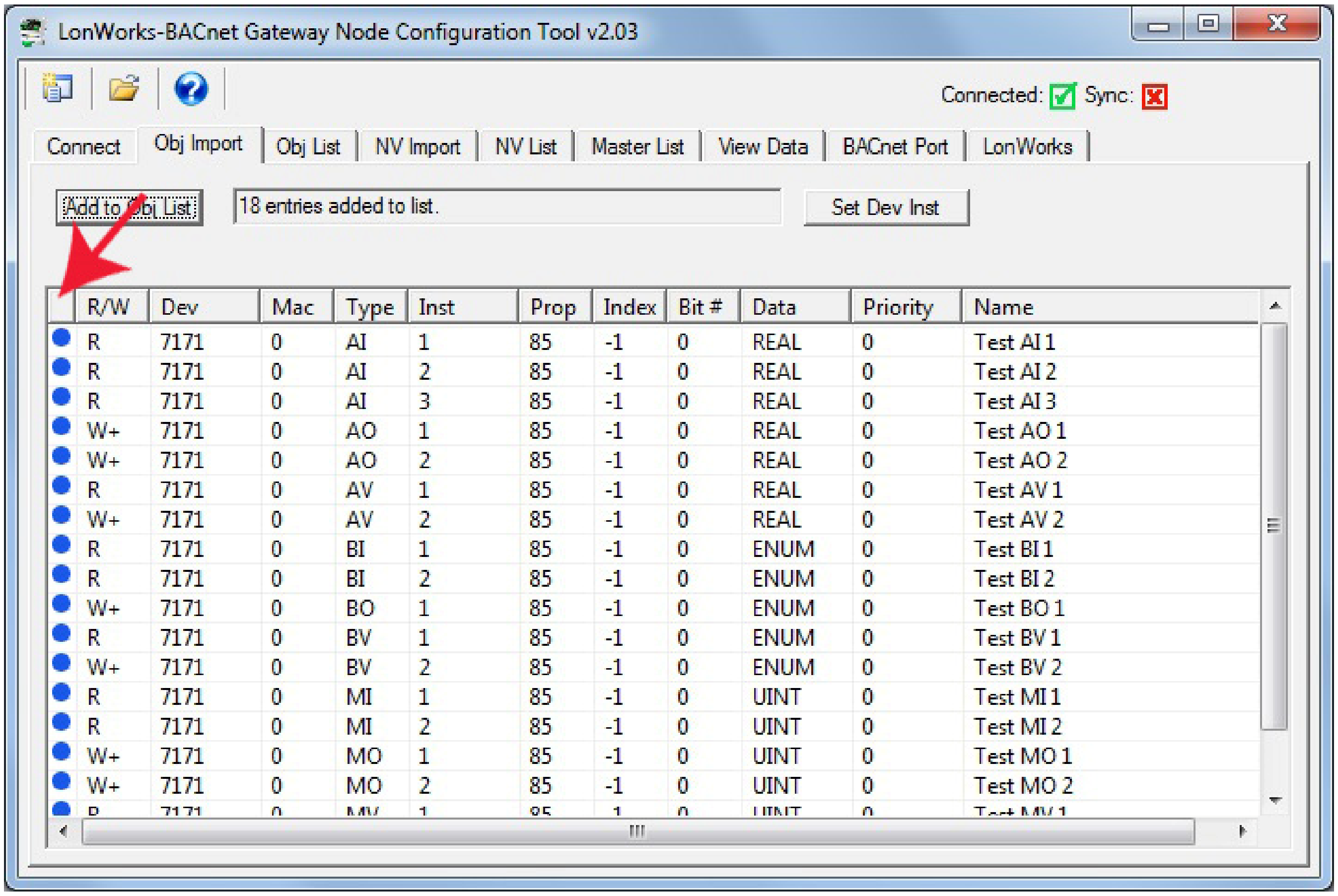
18 objects parsed from objlist3.csv.

	R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
	R	7171	0	AI	1	85	-1	0	REAL	0	Test AI 1
	R	7171	0	AI	2	85	-1	0	REAL	0	Test AI 2
	R	7171	0	AI	3	85	-1	0	REAL	0	Test AI 3
	W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO 1
	W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO 2
	R	7171	0	AV	1	85	-1	0	REAL	0	Test AV 1
	W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV 2
	R	7171	0	BI	1	85	-1	0	ENUM	0	Test BI 1
	R	7171	0	BI	2	85	-1	0	ENUM	0	Test BI 2
	W+	7171	0	BO	1	85	-1	0	ENUM	0	Test BO 1
	R	7171	0	BV	1	85	-1	0	ENUM	0	Test BV 1
	W+	7171	0	BV	2	85	-1	0	ENUM	0	Test BV 2
	R	7171	0	MI	1	85	-1	0	UINT	0	Test MI 1
	R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
	W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
	W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
	R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1

At this point, the object list is contained in what amounts to a "scratch pad". The CSV file is imported into an interim list so that you can choose which objects you want to include in your configuration. If you created the CSV file, you most likely want all of them. But if somebody else sent you a CSV file containing every object in their BACnet device, you will usually want to select only those that are pertinent to your application.

To select all objects, click the icon column header as illustrated by the red arrow below. This will cause the blue dot to appear on each line, which indicates that this object has been selected to be included in the configuration. To select only certain objects, click the icon area of only those rows you wish to include.

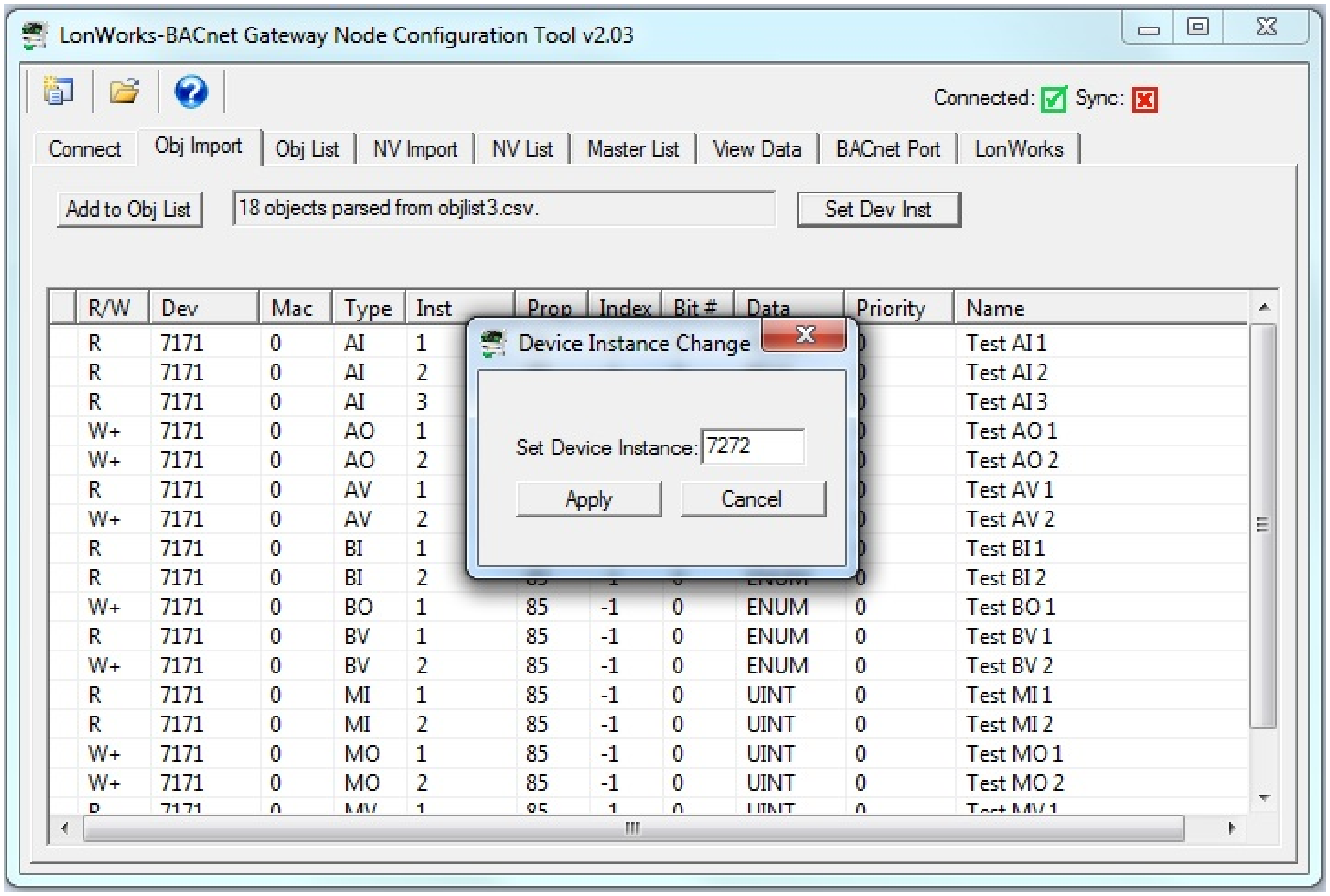
Once you have selected the objects, click the "Add to Obj List" button. Until you click this button, you have no objects in the configured object list. You may, at this point, import another CSV and continue to add multiple objects from multiple CSV files.



18 entries added to list.

	R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
●	R	7171	0	AI	1	85	-1	0	REAL	0	Test AI 1
●	R	7171	0	AI	2	85	-1	0	REAL	0	Test AI 2
●	R	7171	0	AI	3	85	-1	0	REAL	0	Test AI 3
●	W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO 1
●	W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO 2
●	R	7171	0	AV	1	85	-1	0	REAL	0	Test AV 1
●	W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV 2
●	R	7171	0	BI	1	85	-1	0	ENUM	0	Test BI 1
●	R	7171	0	BI	2	85	-1	0	ENUM	0	Test BI 2
●	W+	7171	0	BO	1	85	-1	0	ENUM	0	Test BO 1
●	R	7171	0	BV	1	85	-1	0	ENUM	0	Test BV 1
●	W+	7171	0	BV	2	85	-1	0	ENUM	0	Test BV 2
●	R	7171	0	MI	1	85	-1	0	UINT	0	Test MI 1
●	R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
●	W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
●	W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
●	R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1

If you will be connecting two or more of the same type of BACnet device, each having identical object sets, click "Add to Obj List". Then use "Set Dev Inst" to select the next device's instance (identifier that will be sent in Who-Is message). Now click "Add to Obj List" again. This adds the same set of objects a second time, but with a different device instance the second time.



18 objects parsed from objlist3.csv.

Set Device Instance: 7272

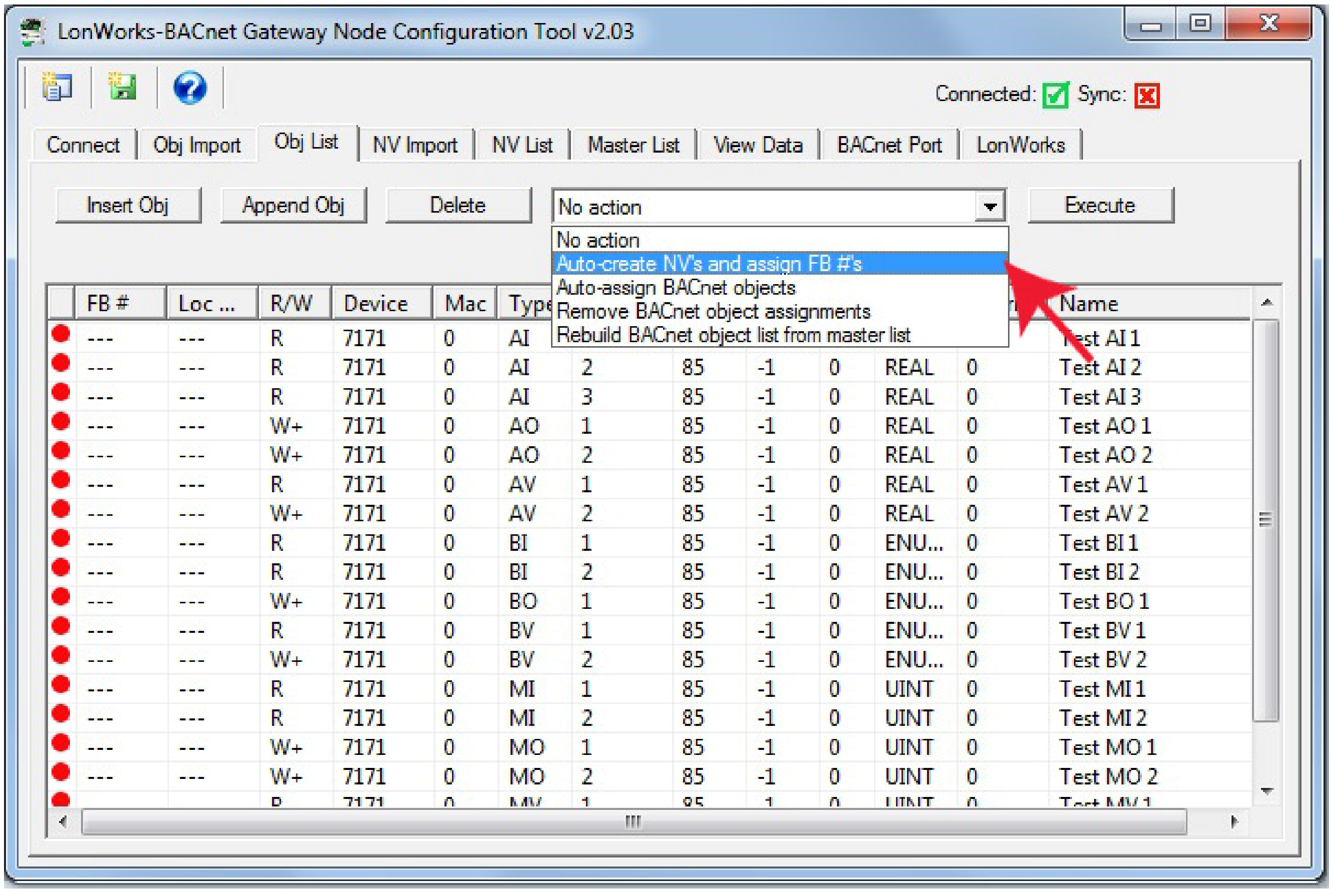
R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
R	7171	0	AI	1						Test AI 1
R	7171	0	AI	2						Test AI 2
R	7171	0	AI	3						Test AI 3
W+	7171	0	AO	1						Test AO 1
W+	7171	0	AO	2						Test AO 2
R	7171	0	AV	1						Test AV 1
W+	7171	0	AV	2						Test AV 2
R	7171	0	BI	1						Test BI 1
R	7171	0	BI	2						Test BI 2
W+	7171	0	BO	1	85	-1	0	ENUM	0	Test BO 1
R	7171	0	BV	1	85	-1	0	ENUM	0	Test BV 1
W+	7171	0	BV	2	85	-1	0	ENUM	0	Test BV 2
R	7171	0	MI	1	85	-1	0	UINT	0	Test MI 1
R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1

The use of the file open icon on the toolbar at the top of the screen is mentioned above. The first icon on the toolbar is the 'new' button. This completely clears the object import list. The other icon is the help button. Click that button any time you want to open a copy of this page.

6 Tool 'Obj List' Page

6.1 Auto-Building the Configuration

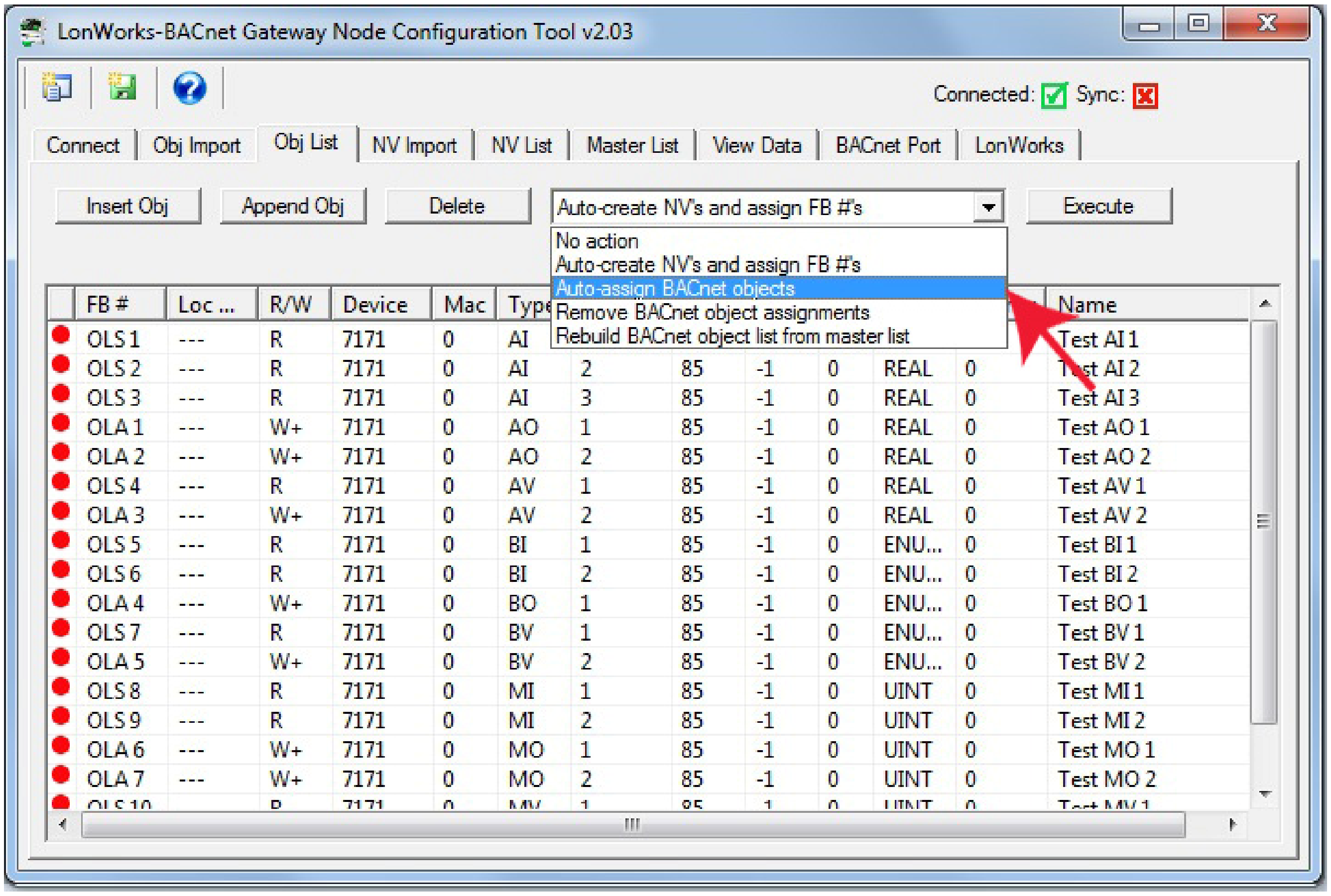
Once you have imported your object list(s) on the Obj Import page, go to the Obj List page. Here is where you begin the auto-build of the rest of your configuration. Start by selecting "Auto-create NV's and assign FB #'s". Then click the Execute button.


LonWorks-BACnet Gateway Node Configuration Tool v2.03
Connected:
Sync:
Connect
Obj Import
Obj List
NV Import
NV List
Master List
View Data
BACnet Port
LonWorks
Insert Obj
Append Obj
Delete
No action
Execute
No action
Auto-create NV's and assign FB #'s
Auto-assign BACnet objects
Remove BACnet object assignments
Rebuild BACnet object list from master list

	FB #	Loc ...	R/W	Device	Mac	Type							Name
●	---	---	R	7171	0	AI							Test AI 1
●	---	---	R	7171	0	AI	2	85	-1	0	REAL	0	Test AI 2
●	---	---	R	7171	0	AI	3	85	-1	0	REAL	0	Test AI 3
●	---	---	W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO 1
●	---	---	W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO 2
●	---	---	R	7171	0	AV	1	85	-1	0	REAL	0	Test AV 1
●	---	---	W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV 2
●	---	---	R	7171	0	BI	1	85	-1	0	ENU...	0	Test BI 1
●	---	---	R	7171	0	BI	2	85	-1	0	ENU...	0	Test BI 2
●	---	---	W+	7171	0	BO	1	85	-1	0	ENU...	0	Test BO 1
●	---	---	R	7171	0	BV	1	85	-1	0	ENU...	0	Test BV 1
●	---	---	W+	7171	0	BV	2	85	-1	0	ENU...	0	Test BV 2
●	---	---	R	7171	0	MI	1	85	-1	0	UINT	0	Test MI 1
●	---	---	R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
●	---	---	W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
●	---	---	W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
●			R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1

Upon executing the Auto-create NV's, the FB# column will be populated. BACnet objects designated as "W" for write, meaning the gateway will write to these objects in another BACnet device (assuming gateway is a client), will be assigned to an Open Loop Actuator function block having a Network Variable Input (NVI). BACnet objects designated as "R" for read, meaning the gateway will read these objects from another BACnet device, will be assigned to an Open Loop Sensor function block having a Network Variable Output (NVO).

Next, select "Auto-assign BACnet objects" from the list and click Execute.


LonWorks-BACnet Gateway Node Configuration Tool v2.03

Connected: ✓ Sync: ✗

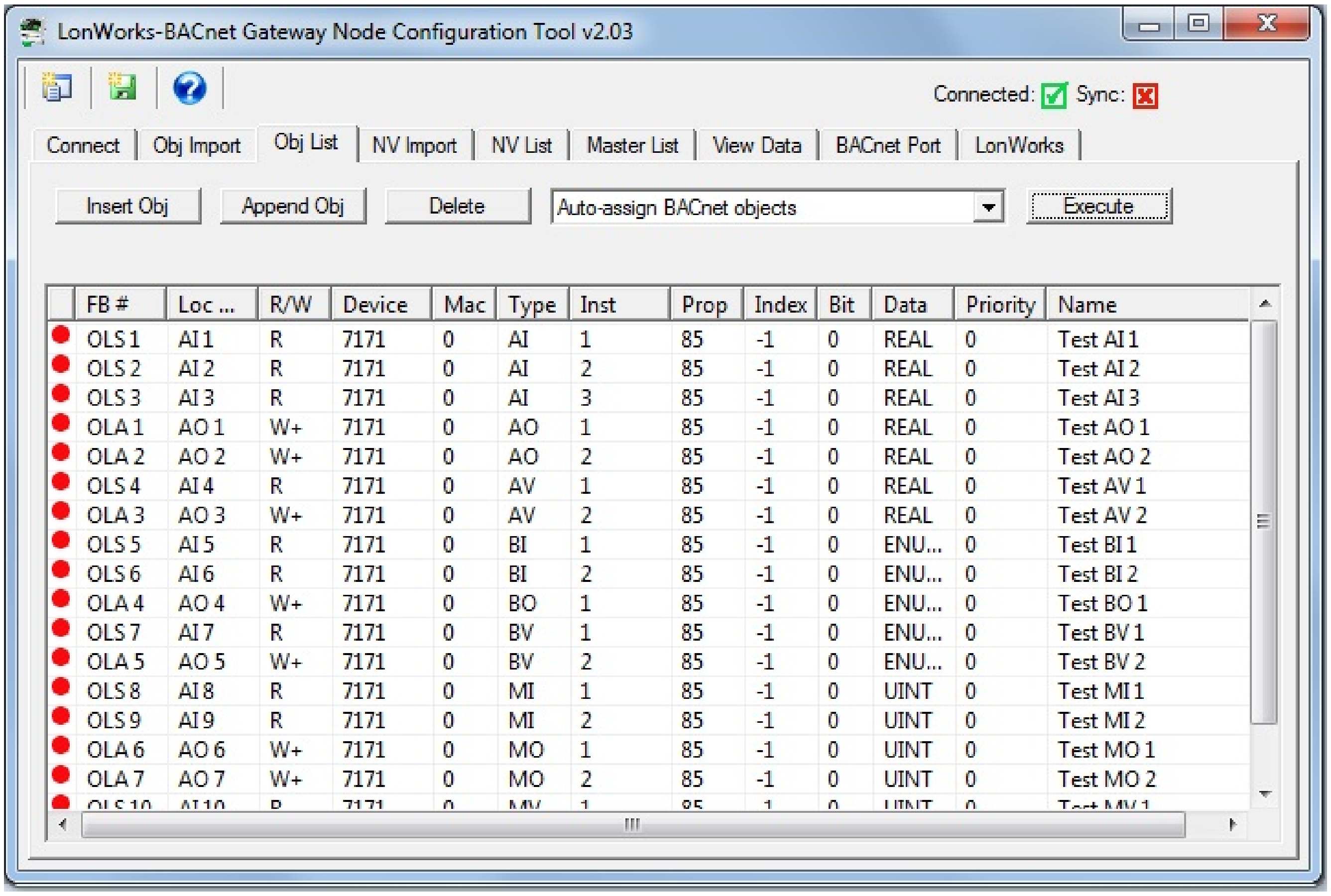
Connect | Obj Import | Obj List | NV Import | NV List | Master List | View Data | BACnet Port | LonWorks

Insert Obj | Append Obj | Delete | Auto-create NV's and assign FB #'s | Execute

No action
Auto-create NV's and assign FB #'s
Auto-assign BACnet objects
Remove BACnet object assignments
Rebuild BACnet object list from master list

	FB #	Loc ...	R/W	Device	Mac	Typ							Name
●	OLS 1	---	R	7171	0	AI							Test AI 1
●	OLS 2	---	R	7171	0	AI	2	85	-1	0	REAL	0	Test AI 2
●	OLS 3	---	R	7171	0	AI	3	85	-1	0	REAL	0	Test AI 3
●	OLA 1	---	W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO 1
●	OLA 2	---	W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO 2
●	OLS 4	---	R	7171	0	AV	1	85	-1	0	REAL	0	Test AV 1
●	OLA 3	---	W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV 2
●	OLS 5	---	R	7171	0	BI	1	85	-1	0	ENU...	0	Test BI 1
●	OLS 6	---	R	7171	0	BI	2	85	-1	0	ENU...	0	Test BI 2
●	OLA 4	---	W+	7171	0	BO	1	85	-1	0	ENU...	0	Test BO 1
●	OLS 7	---	R	7171	0	BV	1	85	-1	0	ENU...	0	Test BV 1
●	OLA 5	---	W+	7171	0	BV	2	85	-1	0	ENU...	0	Test BV 2
●	OLS 8	---	R	7171	0	MI	1	85	-1	0	UINT	0	Test MI 1
●	OLS 9	---	R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
●	OLA 6	---	W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
●	OLA 7	---	W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
●	OLS 10		R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1

Upon executing Auto-assign BACnet objects, the object numbers allocated will appear in the Loc... (Local Object) column.


LonWorks-BACnet Gateway Node Configuration Tool v2.03

Connected: ✓ Sync: ✗

Connect | Obj Import | Obj List | NV Import | NV List | Master List | View Data | BACnet Port | LonWorks

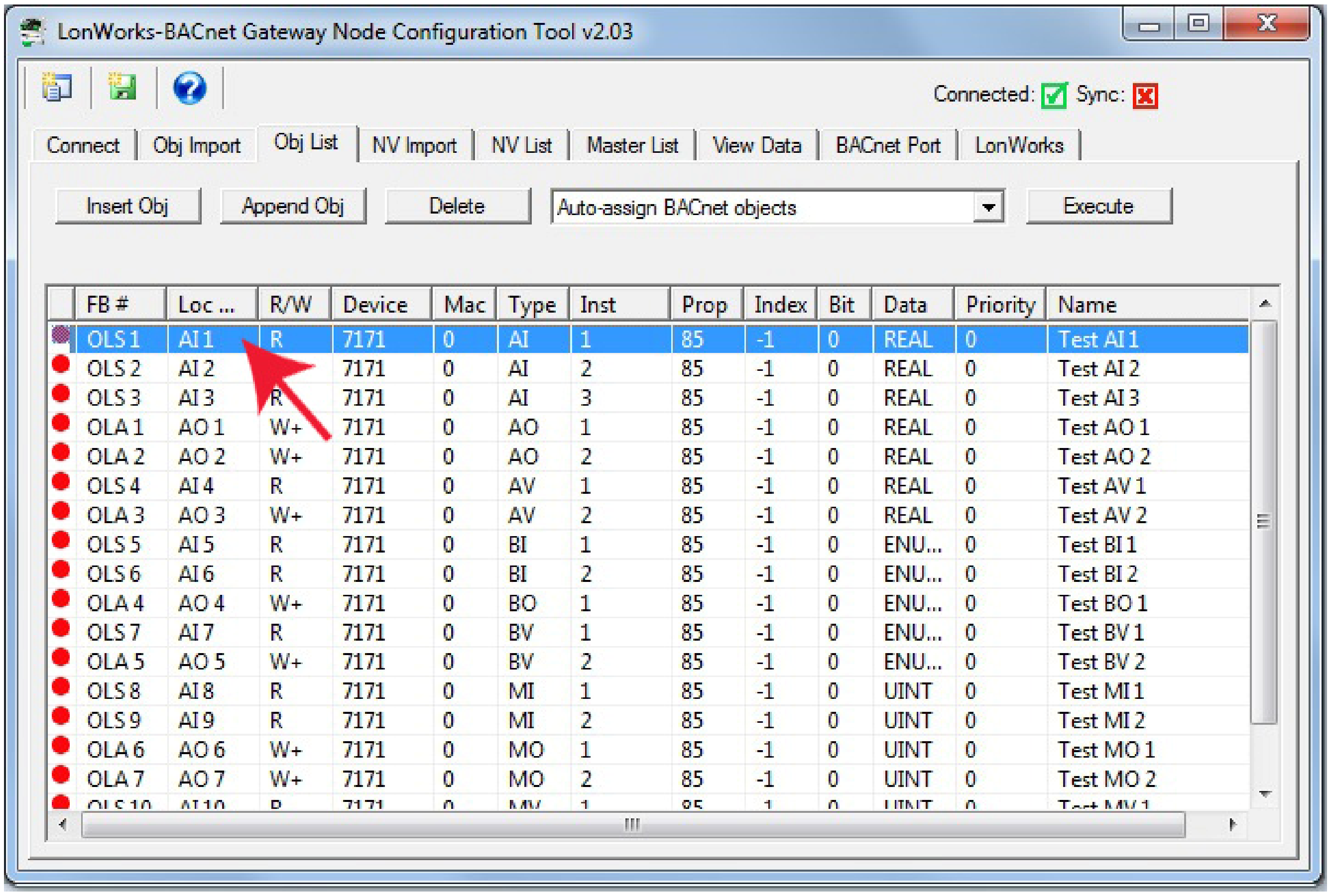
Insert Obj | Append Obj | Delete | Auto-assign BACnet objects | Execute

	FB #	Loc ...	R/W	Device	Mac	Type	Inst	Prop	Index	Bit	Data	Priority	Name
●	OLS 1	AI 1	R	7171	0	AI	1	85	-1	0	REAL	0	Test AI 1
●	OLS 2	AI 2	R	7171	0	AI	2	85	-1	0	REAL	0	Test AI 2
●	OLS 3	AI 3	R	7171	0	AI	3	85	-1	0	REAL	0	Test AI 3
●	OLA 1	AO 1	W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO 1
●	OLA 2	AO 2	W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO 2
●	OLS 4	AI 4	R	7171	0	AV	1	85	-1	0	REAL	0	Test AV 1
●	OLA 3	AO 3	W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV 2
●	OLS 5	AI 5	R	7171	0	BI	1	85	-1	0	ENU...	0	Test BI 1
●	OLS 6	AI 6	R	7171	0	BI	2	85	-1	0	ENU...	0	Test BI 2
●	OLA 4	AO 4	W+	7171	0	BO	1	85	-1	0	ENU...	0	Test BO 1
●	OLS 7	AI 7	R	7171	0	BV	1	85	-1	0	ENU...	0	Test BV 1
●	OLA 5	AO 5	W+	7171	0	BV	2	85	-1	0	ENU...	0	Test BV 2
●	OLS 8	AI 8	R	7171	0	MI	1	85	-1	0	UINT	0	Test MI 1
●	OLS 9	AI 9	R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
●	OLA 6	AO 6	W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
●	OLA 7	AO 7	W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
●	OLS 10	AI 10	R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1

Creating of the configuration is now largely complete. You can make alterations to things like data scaling at this point if you wish. You can also change network variable types at this point. But for simply putting your BACnet device on the LonWorks network using generic counts as the network variable type, the configuration is complete. All that remains now is to send the configuration to the gateway device. At this point, the configuration only exists on your PC. It needs to get written into the device before the gateway will be functional.

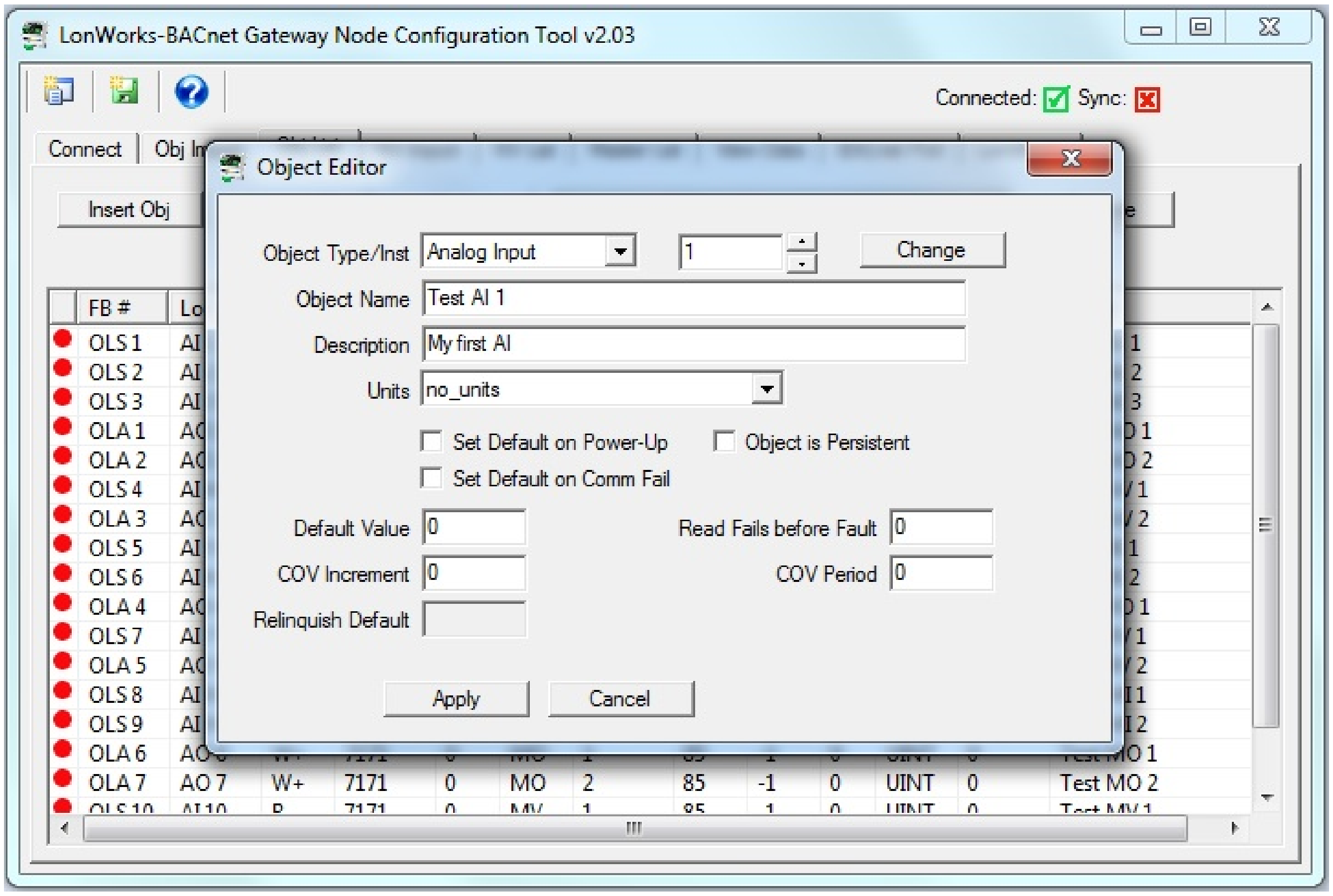
6.2 Editing the Object List

To edit the configuration of an existing BACnet object, double click on that line on the Obj List. There are two possible dialog boxes that will pop up. Clicking a row in the first two columns will open the dialog for editing the local BACnet object. Clicking the remaining columns will open the dialog for editing the map that determines what BACnet object you will read or write in another BACnet device (assuming the gateway will be acting as a client).



	FB #	Loc ...	R/W	Device	Mac	Type	Inst	Prop	Index	Bit	Data	Priority	Name
●	OLS 1	AI 1	R	7171	0	AI	1	85	-1	0	REAL	0	Test AI 1
●	OLS 2	AI 2		7171	0	AI	2	85	-1	0	REAL	0	Test AI 2
●	OLS 3	AI 3	R	7171	0	AI	3	85	-1	0	REAL	0	Test AI 3
●	OLA 1	AO 1	W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO 1
●	OLA 2	AO 2	W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO 2
●	OLS 4	AI 4	R	7171	0	AV	1	85	-1	0	REAL	0	Test AV 1
●	OLA 3	AO 3	W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV 2
●	OLS 5	AI 5	R	7171	0	BI	1	85	-1	0	ENU...	0	Test BI 1
●	OLS 6	AI 6	R	7171	0	BI	2	85	-1	0	ENU...	0	Test BI 2
●	OLA 4	AO 4	W+	7171	0	BO	1	85	-1	0	ENU...	0	Test BO 1
●	OLS 7	AI 7	R	7171	0	BV	1	85	-1	0	ENU...	0	Test BV 1
●	OLA 5	AO 5	W+	7171	0	BV	2	85	-1	0	ENU...	0	Test BV 2
●	OLS 8	AI 8	R	7171	0	MI	1	85	-1	0	UINT	0	Test MI 1
●	OLS 9	AI 9	R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
●	OLA 6	AO 6	W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
●	OLA 7	AO 7	W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
●	OLS 10	AI 10	R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1

Upon double clicking the first couple of columns in a line on the Obj List, the dialog shown below will pop up for editing the local BACnet object.



Object Type/Inst: Analog Input, 1 [Change]

Object Name: Test AI 1

Description: My first AI

Units: no_units

☐ Set Default on Power-Up ☐ Object is Persistent

☐ Set Default on Comm Fail

Default Value: 0 Read Fails before Fault: 0

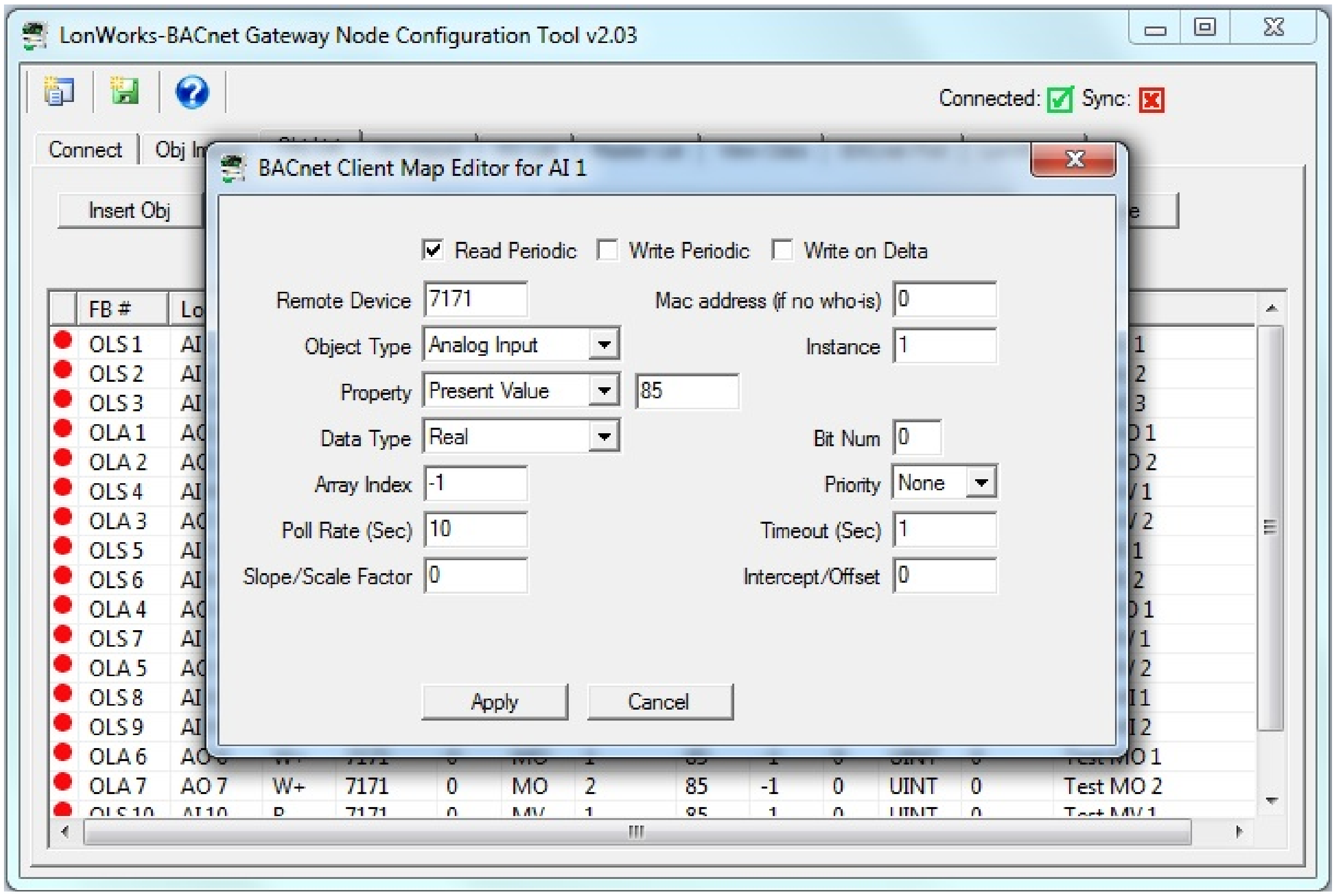
COV Increment: 0 COV Period: 0

Relinquish Default:

FB #	Lo
OLS 1	AI
OLS 2	AI
OLS 3	AI
OLA 1	AO
OLA 2	AO
OLS 4	AI
OLA 3	AO
OLS 5	AI
OLS 6	AI
OLA 4	AO
OLS 7	AI
OLA 5	AO
OLS 8	AI
OLS 9	AI
OLA 6	AO
OLA 7	AO 7

OLA 7 AO 7 W+ 7171 0 MO 2 85 -1 0 UINT 0 Test MO 2

Upon double clicking the last several columns in a line on the Obj List, the dialog shown below will pop up for editing the mapping to a remote BACnet object that the gateway will read or write when acting as a BACnet client. The first example is for reading an Analog Input from another BACnet device.


LonWorks-BACnet Gateway Node Configuration Tool v2.03

Connected: ✓ Sync: ✗

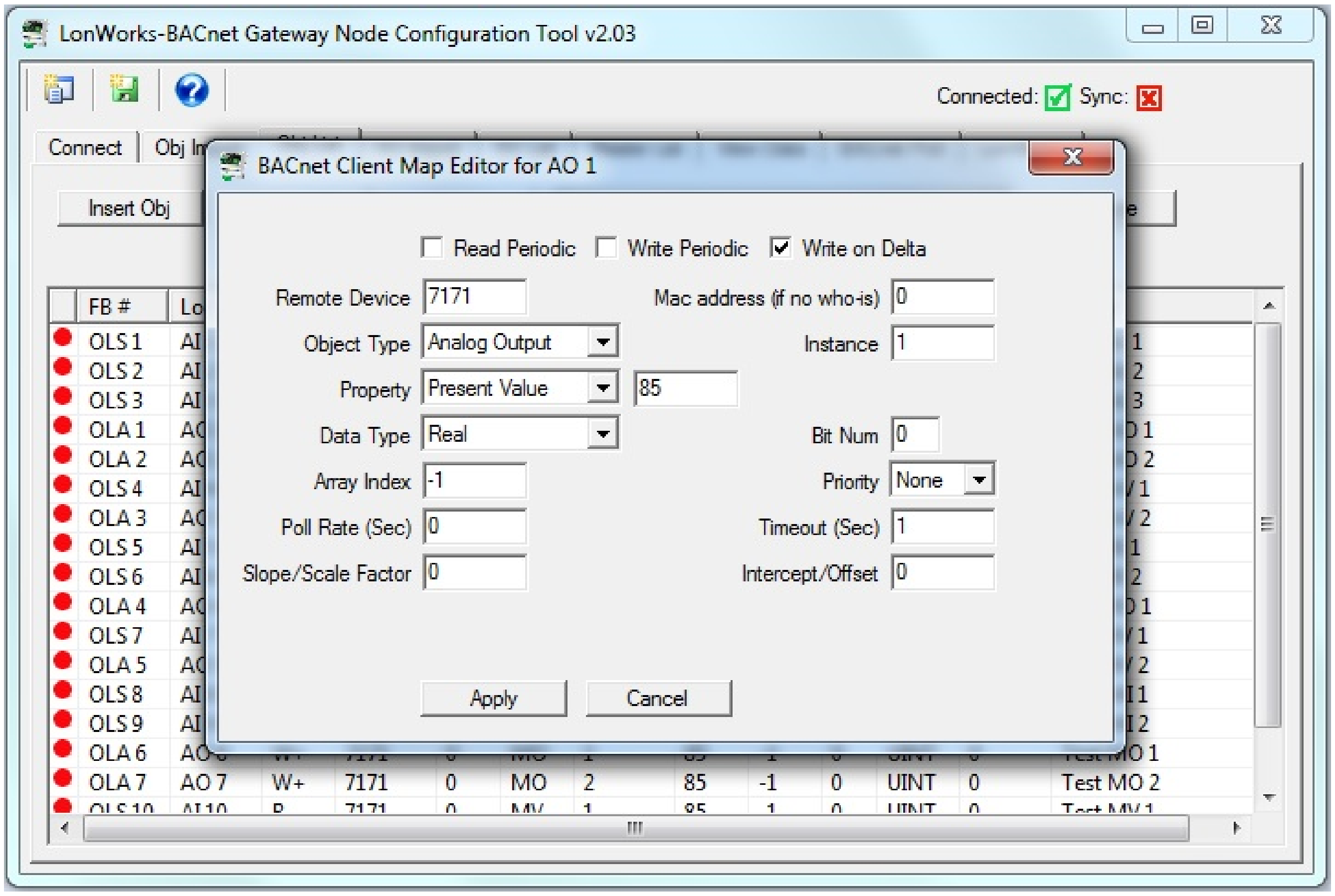
BACnet Client Map Editor for AI 1

[✓] Read Periodic [ ] Write Periodic [ ] Write on Delta

Remote Device	7171	Mac address (if no who-is)	0
Object Type	Analog Input	Instance	1
Property	Present Value 85		
Data Type	Real	Bit Num	0
Array Index	-1	Priority	None
Poll Rate (Sec)	10	Timeout (Sec)	1
Slope/Scale Factor	0	Intercept/Offset	0

	FB #	Lo											
●	OLS 1	AI											1
●	OLS 2	AI											2
●	OLS 3	AI											3
●	OLA 1	AO											O 1
●	OLA 2	AO											O 2
●	OLS 4	AI											V 1
●	OLA 3	AO											V 2
●	OLS 5	AI											1
●	OLS 6	AI											2
●	OLA 4	AO											O 1
●	OLS 7	AI											V 1
●	OLA 5	AO											V 2
●	OLS 8	AI											I 1
●	OLS 9	AI											I 2
●	OLA 6	AO 6	W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
●	OLA 7	AO 7	W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
●	OLS 10	AI 10	R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1

The second example (below) is for writing to an Analog Output in another BACnet device.



BACnet Client Map Editor for AO 1

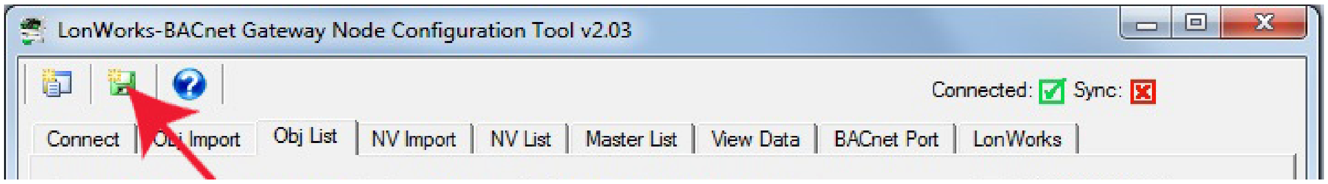
[ ] Read Periodic [ ] Write Periodic [✓] Write on Delta

Remote Device	7171	Mac address (if no who-is)	0
Object Type	Analog Output	Instance	1
Property	Present Value		85
Data Type	Real	Bit Num	0
Array Index	-1	Priority	None
Poll Rate (Sec)	0	Timeout (Sec)	1
Slope/Scale Factor	0	Intercept/Offset	0

Apply Cancel

6.3 Object List Export

Once you have created a BACnet object list, you can export your object list.

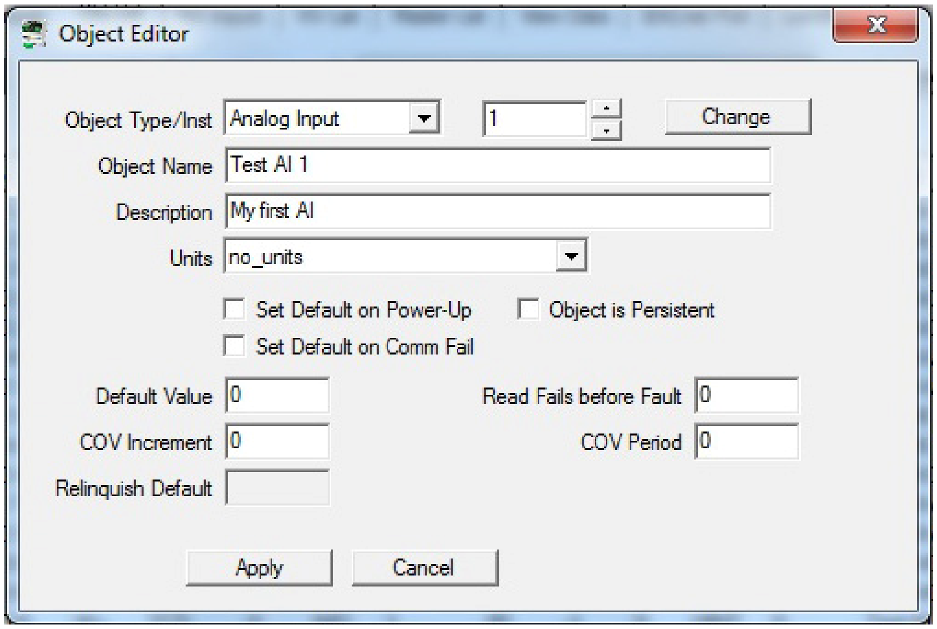


To export, click the new file icon on the toolbar. The CSV file that is exported will be in the file format described in Appendix D of this user guide. This means you can import the same file back into the configuration tool (on Obj Import page) at a later time, or open the file using a spreadsheet program to create documentation of your BACnet object set.

6.4 Definition of BACnet Object Configuration Parameters

The BACnet Object Editor dialog is accessed by clicking the first couple of columns on a line on the Obj List page, or by clicking the local object portion of a line on the Master List page.

The Object Editor dialog applies to BACnet objects that are contained within the Babel Buster gateway. Other BACnet clients can read and write standard BACnet properties of these objects.



Object Type/Inst: Analog Input, 1
Object Name: Test AI 1
Description: My first AI
Units: no_units
Set Default on Power-Up [ ]
Object is Persistent [ ]
Set Default on Comm Fail [ ]
Default Value: 0
Read Fails before Fault: 0
COV Increment: 0
COV Period: 0
Relinquish Default:

The "Change" button is used to initiate change in object type and/or instance number. It is possible to change any object from Analog to Binary, for example. It is also possible to change an object from input to output. Changes in direction (input vs. output) should be done before assigning network variables so that the corresponding network variable also ends up going the correct direction. To change an objec type or instance, select the desired new values, and click Change. This will test your selection to see if it conflicts with other objects that have already been allocated. If the change is unacceptable, a pop-up messate will indicat that. Otherwise, the change will become effective upon clicking the Apply button at the bottom of this dialog. If you close the dialog without clicking Apply, or you click Cancel instead, the change in object type/instance will not be executed.

The parameters found in the Object Editor dialog are as follows.

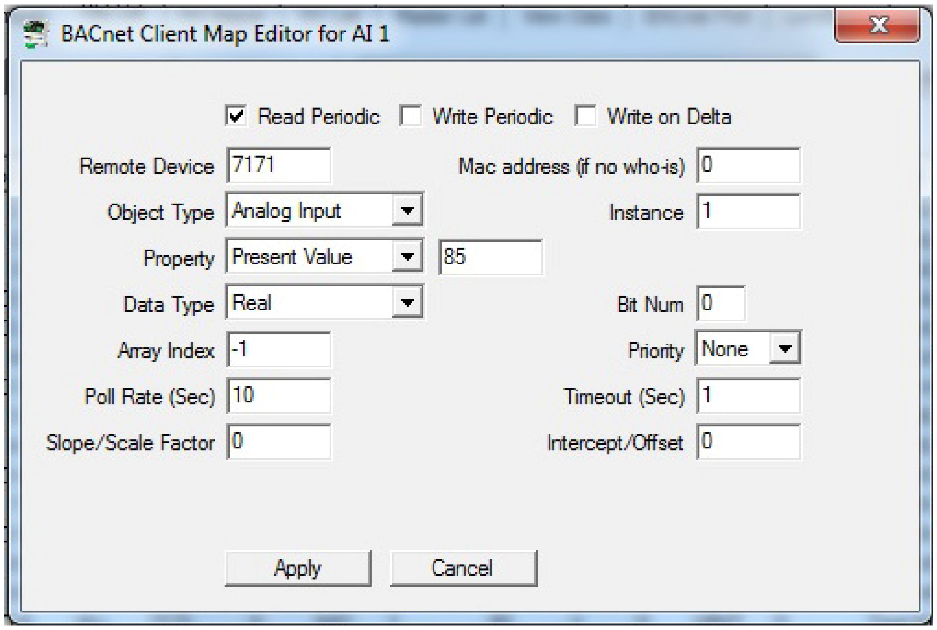
Object Type/Inst	Select BACnet object type from the list, and enter or scroll to the desired object instance. The object number will stop scrolling at the limit of the configuration tool, but this instance could still exceed the object limit of the device - you will check device instance limits on the BACnet Port page.
Object Name	Enter an alphanumeric name for the object. The name must be unique, and will be checked for uniqueness when you click Apply. If it is a duplicate name, you will see an error message and the dialog will remain open.
Description	Enter any arbitrary description for the object. The description is not required to be unique.
Units	Select a BACnet units type from the list.
Set Default on Power-Up	Check this box if the object should assume the default value every time the BB2-3020/7020 powers up.
Object is Persistent	Check this box if this object should retain the most recently received value after a restart or power cycle. If this option is not selected, the object will default to a value of zero until new data is received from some source.
Set Default on Comm Fail	Check this box if the object should assume the default value when communication with the slave device has failed some number of times.
Default Value	This is the value that should become the Present Value upon power-up or upon communications failure, if either of these options are selected by the

	appropriate check boxes above.
Read Fails before Fault	When 'Set Default on Comm Fail' is checked, this entry will allow you to disregard a small number of spurious errors. At least this many errors must occur consecutively before a fault will actually be flagged. Causing a fault as a result of a single instance of spurious noise on a communication line can be a nuisance. This setting allows quieting the nuisance fault notifications.
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.
COV Period	The initial COV Period will take effect when the Babel Buster 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.
Relinquish Default	The initial Relinquish Default will take effect when the gateway is first powered up. After that, any BACnet client can write a new Relinquish Default value to the object. The Relinquish Default value applies to any Commandable object, and the Present Value of the object will assume this value when all 16 priorities have been relinquished.

6.5 Definition of BACnet Client Map Configuration Parameters

The BACnet Client Map Editor dialog is accessed by clicking the last several columns on a line on the Obj List page, or by clicking the map portion of a line on the Master List page.

The BACnet Client Map Editor defines the object that will be read from or written to a remote BACnet server (slave) device when the Babel Buster is acting as a BACnet client.


BACnet Client Map Editor for AI 1

[✓] Read Periodic [ ] Write Periodic [ ] Write on Delta

Remote Device: 7171
Mac address (if no who-is): 0
Object Type: Analog Input
Instance: 1
Property: Present Value 85
Data Type: Real
Bit Num: 0
Array Index: -1
Priority: None
Poll Rate (Sec): 10
Timeout (Sec): 1
Slope/Scale Factor: 0
Intercept/Offset: 0

The parameters found in the BACnet Client Map Editor dialog are as follows.

Read Periodic	Tells the gateway to read this object (from another BACnet device) at the rate set by the Poll Rate. Only valid for Open Loop Sensor function blocks.
Write Periodic	Tells the gateway (master) to write this object at the rate set by the Poll Rate. Only valid for Open Loop Actuator function blocks.
Write On Delta	Tells the gateway to write this object when new data is received from LonWorks. One or the other of the 'write' options can be selected, but not both. Only valid for Open Loop Actuator function blocks.
Remote Device	Enter the device instance of the remote BACnet server (slave). The remote device must be able to support Who-Is/I-Am for locating device by device instance.
Mac address (if no who-is)	Sets the MS/TP Mac address (station ID) if the slave device does not support Who-Is and I-Am. Device instance will be used to automatically capture the Mac address when Who-Is/I-Am is supported. This setting should be zero to use Who-Is, or set to some non-zero Mac address if the slave does not support Who-Is. This setting applies to MS/TP only. BACnet IP devices are required to support Who-Is.
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.
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'.
Data Type	Select the data type expected from this list. Note that some types are not supported here. Character string, for example, is one type of data that cannot be forced into Analog, Binary, or Multi-state objects by the BB2-3020 or BB2-7020.
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.
Array Index	Some properties are an array of values. If applicable, enter the array index here. Enter -1 to indicate no array index is to be used. Zero is a valid index.
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 (Sec)	When reading or writing periodically, set this to the number of seconds between polls.
Timeout (Sec)	This defines the amount of time that the Babel Buster, acting as BACnet client, will wait for a response from the other BACnet device before calling it an error.
Slope/Scale Factor	Data is multiplied by this value when read from another BACnet device, before storing to the internal gateway data object (and hence LonWorks NV). The process is reversed when writing to another BACnet device. A scale value of zero will mean 'no scaling' which would be mathematically equivalent to scale=1.0.

Intercept/Offset	This value is added to data when read from another BACnet device, before storing to the internal gateway data object (and hence LonWorks NV). The process is reversed when writing to another BACnet device. Offset is applied after scale when reading, and vice versa when writing.

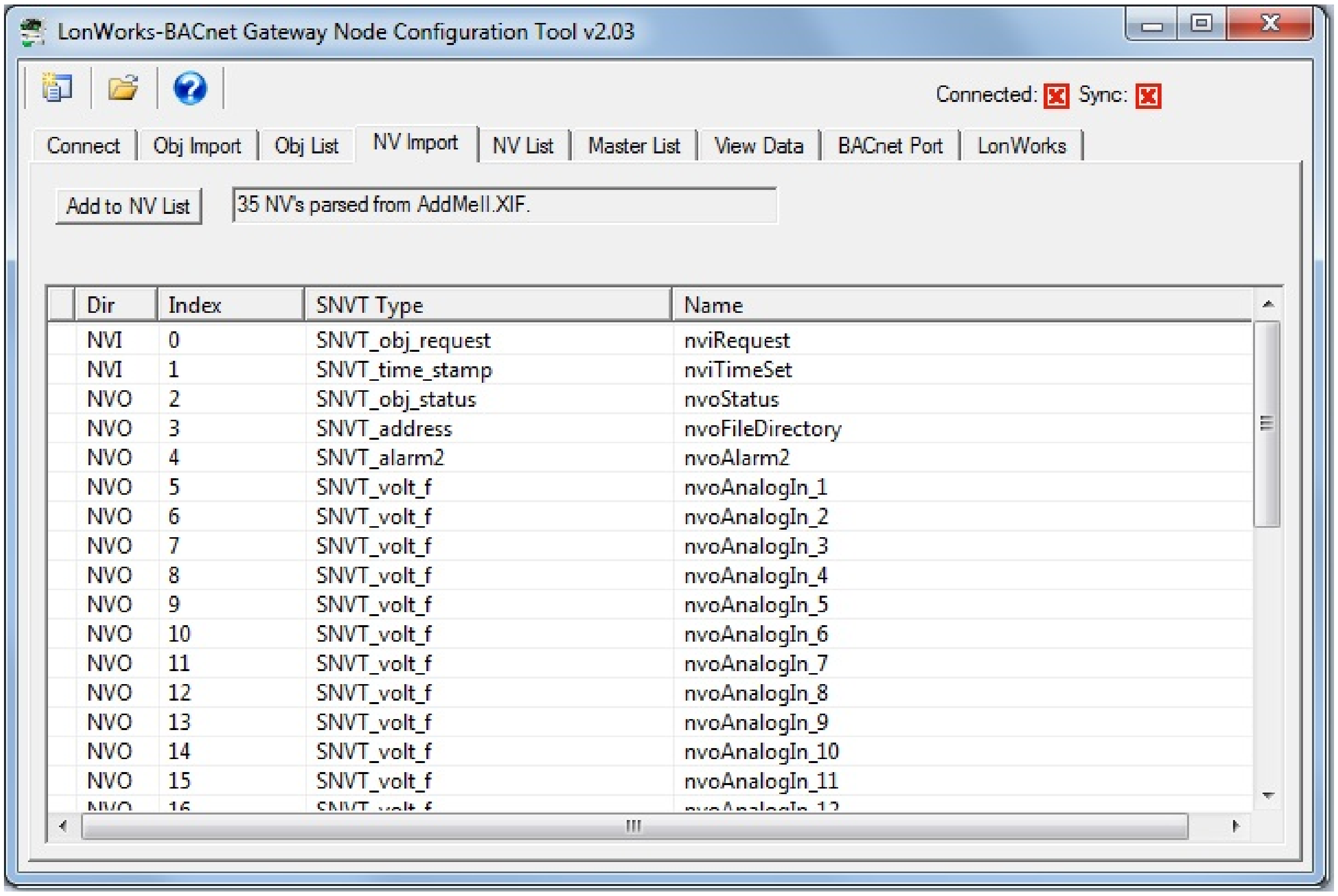
7 Tool 'NV Import' Page

7.1 Importing an XIF File

Starting from an XIF file for a specific LonWorks device, and auto-building the BACnet object list, is the approach you would take if you want to make a BACnet device mimic that particular LonWorks device on the LonWorks network.

IMPORTANT: If your intent is to put a LonWorks device on a BACnet network, this is not the correct gateway to use. The gateway discussed here only applies where you are interfacing BACnet devices to a LonWorks network, and the LonWorks network is the primary network of the overall system.

To begin building your configuration from an XIF file, go to the NV Import page. Click on the file icon to open an XIF file. Once the XIF is imported, the list of network variables will be displayed on the NV Import page.

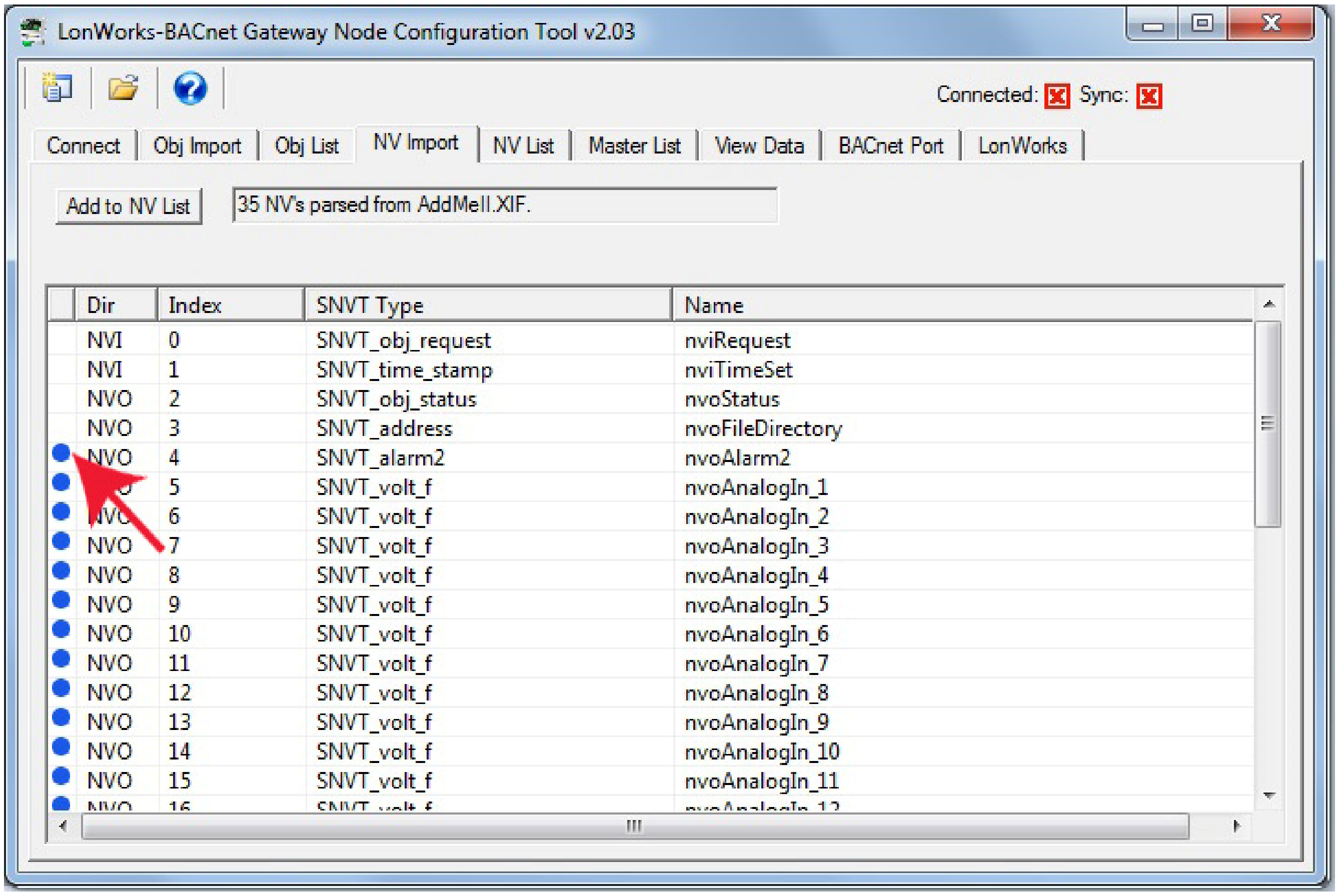


35 NV's parsed from AddMeII.XIF.

Dir	Index	SNVT Type	Name
NVI	0	SNVT_obj_request	nviRequest
NVI	1	SNVT_time_stamp	nviTimeSet
NVO	2	SNVT_obj_status	nvoStatus
NVO	3	SNVT_address	nvoFileDirectory
NVO	4	SNVT_alarm2	nvoAlarm2
NVO	5	SNVT_volt_f	nvoAnalogIn_1
NVO	6	SNVT_volt_f	nvoAnalogIn_2
NVO	7	SNVT_volt_f	nvoAnalogIn_3
NVO	8	SNVT_volt_f	nvoAnalogIn_4
NVO	9	SNVT_volt_f	nvoAnalogIn_5
NVO	10	SNVT_volt_f	nvoAnalogIn_6
NVO	11	SNVT_volt_f	nvoAnalogIn_7
NVO	12	SNVT_volt_f	nvoAnalogIn_8
NVO	13	SNVT_volt_f	nvoAnalogIn_9
NVO	14	SNVT_volt_f	nvoAnalogIn_10
NVO	15	SNVT_volt_f	nvoAnalogIn_11
NVO	16	SNVT_volt_f	nvoAnalogIn_12

The NV Import page is essentially a scratch pad where you import the content of XIF files, then select which of the available variables you wish to include in your gateway configuration. Click on the icon column header to select all items, or click on the icon column for individual lines to select only those lines. The icon will show a blue dot for those lines that are about to be included.

When you have made your selections, click Add to NV List. The selected variables are now copied to the NV List.



35 NV's parsed from AddMeII.XIF.

	Dir	Index	SNVT Type	Name
	NVI	0	SNVT_obj_request	nviRequest
	NVI	1	SNVT_time_stamp	nviTimeSet
	NVO	2	SNVT_obj_status	nvoStatus
	NVO	3	SNVT_address	nvoFileDirectory
●	NVO	4	SNVT_alarm2	nvoAlarm2
●	[illegible]	5	SNVT_volt_f	nvoAnalogIn_1
●	[illegible]	6	SNVT_volt_f	nvoAnalogIn_2
●	NVO	7	SNVT_volt_f	nvoAnalogIn_3
●	NVO	8	SNVT_volt_f	nvoAnalogIn_4
●	NVO	9	SNVT_volt_f	nvoAnalogIn_5
●	NVO	10	SNVT_volt_f	nvoAnalogIn_6
●	NVO	11	SNVT_volt_f	nvoAnalogIn_7
●	NVO	12	SNVT_volt_f	nvoAnalogIn_8
●	NVO	13	SNVT_volt_f	nvoAnalogIn_9
●	NVO	14	SNVT_volt_f	nvoAnalogIn_10
●	NVO	15	SNVT_volt_f	nvoAnalogIn_11
●	NVO	16	SNVT_volt_f	nvoAnalogIn_12

If you need multiple copies of the same set of network variables, you can click the Add to NV List more than once. The same set of selected NV's will be added again each time you click the button.

You may also import different XIF files and continue to add to the NV List by opening a file, selecting NV's, clicking Add, and repeating the process.

The NV Import list will hold up to 1000 network variables at one time. Most LonWorks devices will have a list much shorter than that, and can potentially be added multiple times. It should be noted however, that a single NV on the NV Import list will turn into multiple entries in the NV List if it is a structured NV. The NV List has a capacity for 1000 entries - which may or may not equate to 1000 network variables depending on whether any of them are structured.

Although the maximum list size is set at 1000, the NV List is will never actually contain 1000 network variables. The LonWorks FT5000 microserver provides 120 network variable inputs, and 120 network variable outputs, for a total of 240 network variables which can be structured variables. If structured, the NV will map to multiple data objects. There is a pool of 300 data objects to work with. Each data object, in turn, maps to a BACnet object.

Therefore, the actual limits will be the lesser of 240 LonWorks network variables or 300 BACnet objects. Furthermore, there is a limit of 120 network variables in each direction, and the direction cannot be changed. This restriction is imposed by the LonWorks architecture.

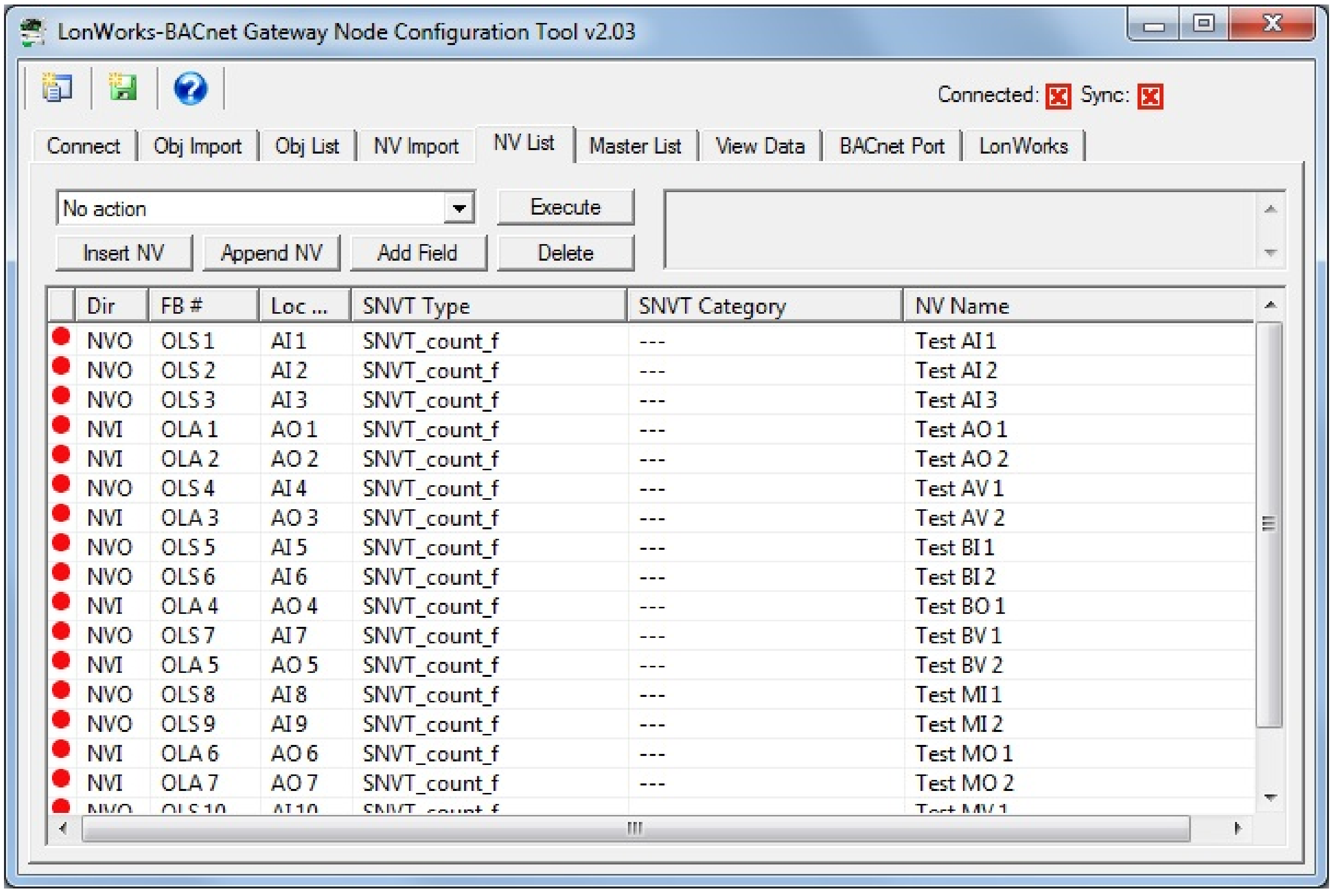
8 Tool 'NV List' Page

8.1 Configuration from Object List

Creating of the configuration is completed primarily on the Obj Import and Obj List pages. Once objects are imported, you come here to the NV List page.

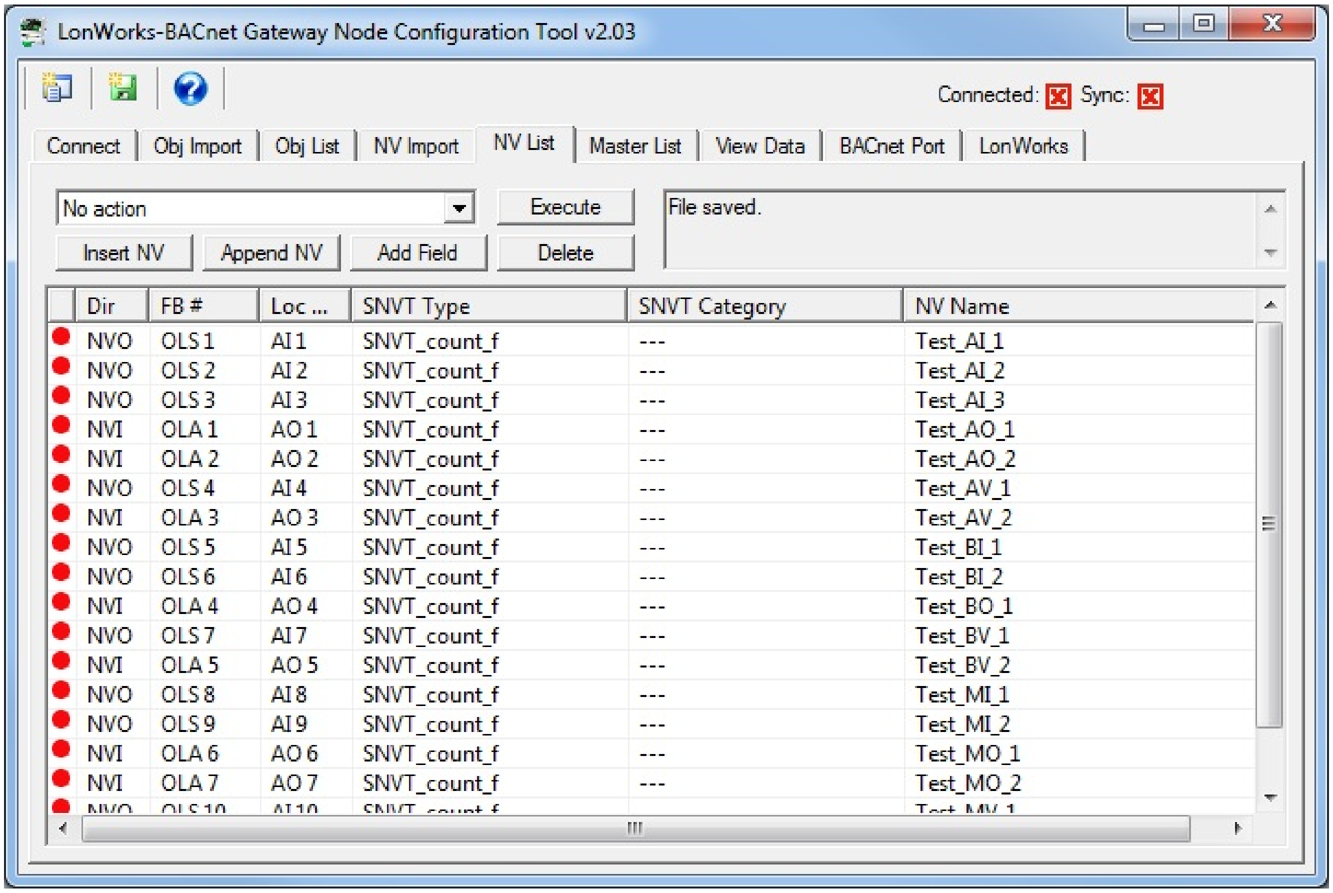
This is the set of network variables that were automatically created for you. If you wish to export an XIF file defining your LonWorks device, you may do so here by clicking the green file icon to create a new file. Some network management tools such as LonMaker can import the XIF directly from the device and you do not need an XIF file. However, some network management tools do require the XIF file, and this is where you create it.

If you want your LonWorks interface to have network variable types other than those that were selected by default, this is where you make those changes.



	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVO	OLS 1	AI 1	SNVT_count_f	---	Test AI 1
●	NVO	OLS 2	AI 2	SNVT_count_f	---	Test AI 2
●	NVO	OLS 3	AI 3	SNVT_count_f	---	Test AI 3
●	NVI	OLA 1	AO 1	SNVT_count_f	---	Test AO 1
●	NVI	OLA 2	AO 2	SNVT_count_f	---	Test AO 2
●	NVO	OLS 4	AI 4	SNVT_count_f	---	Test AV 1
●	NVI	OLA 3	AO 3	SNVT_count_f	---	Test AV 2
●	NVO	OLS 5	AI 5	SNVT_count_f	---	Test BI 1
●	NVO	OLS 6	AI 6	SNVT_count_f	---	Test BI 2
●	NVI	OLA 4	AO 4	SNVT_count_f	---	Test BO 1
●	NVO	OLS 7	AI 7	SNVT_count_f	---	Test BV 1
●	NVI	OLA 5	AO 5	SNVT_count_f	---	Test BV 2
●	NVO	OLS 8	AI 8	SNVT_count_f	---	Test MI 1
●	NVO	OLS 9	AI 9	SNVT_count_f	---	Test MI 2
●	NVI	OLA 6	AO 6	SNVT_count_f	---	Test MO 1
●	NVI	OLA 7	AO 7	SNVT_count_f	---	Test MO 2
●	NVO	OLS 10	AI 10	SNVT_count_f		Test MV 1

The names provided in the CSV file you imported might not necessarily qualify as legal network variable names. A "legal" network variable name is limited to 16 characters and no spaces. Therefore, spaces will be replaced with underscores and names will be truncated in the process of exporting the XIF file. Following XIF export (file save), the NV List above will appear as shown below instead. You may wish to edit the names *before* creating an XIF file.

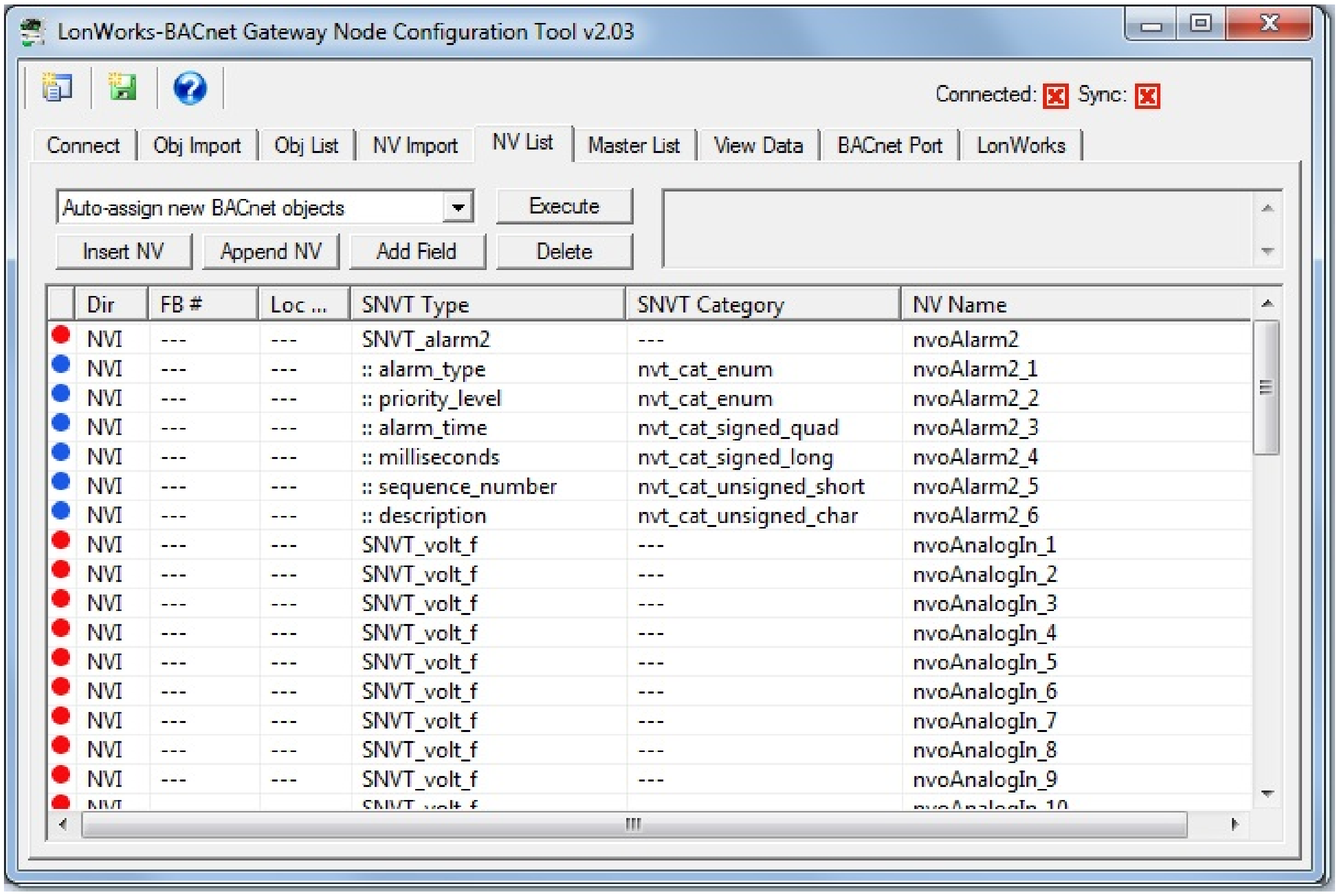


File saved.

	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVO	OLS 1	AI 1	SNVT_count_f	---	Test_AI_1
●	NVO	OLS 2	AI 2	SNVT_count_f	---	Test_AI_2
●	NVO	OLS 3	AI 3	SNVT_count_f	---	Test_AI_3
●	NVI	OLA 1	AO 1	SNVT_count_f	---	Test_AO_1
●	NVI	OLA 2	AO 2	SNVT_count_f	---	Test_AO_2
●	NVO	OLS 4	AI 4	SNVT_count_f	---	Test_AV_1
●	NVI	OLA 3	AO 3	SNVT_count_f	---	Test_AV_2
●	NVO	OLS 5	AI 5	SNVT_count_f	---	Test_BI_1
●	NVO	OLS 6	AI 6	SNVT_count_f	---	Test_BI_2
●	NVI	OLA 4	AO 4	SNVT_count_f	---	Test_BO_1
●	NVO	OLS 7	AI 7	SNVT_count_f	---	Test_BV_1
●	NVI	OLA 5	AO 5	SNVT_count_f	---	Test_BV_2
●	NVO	OLS 8	AI 8	SNVT_count_f	---	Test_MI_1
●	NVO	OLS 9	AI 9	SNVT_count_f	---	Test_MI_2
●	NVI	OLA 6	AO 6	SNVT_count_f	---	Test_MO_1
●	NVI	OLA 7	AO 7	SNVT_count_f	---	Test_MO_2
●	NVO	OLS 10	AI 10	SNVT_count_f		Test_MV_1

8.2 Configuration from XIF File

The NV List is the definition of the list of network variables that will become present in your LonWorks device (i.e. this gateway) when viewed as a node on the LonWorks network. The icon in the first column will be red if this NV definition has not yet been written to the gateway, and green if it has been written to the gateway. Blue icons indicate fields of structured network variables. All lines with a blue icon are part of the network variable immediately preceding the set of blue icons. There will be only one network variable, but multiple BACnet objects, for a structured network variable (refer to previous section in this user guide if you did not already review treatment of structured variables).

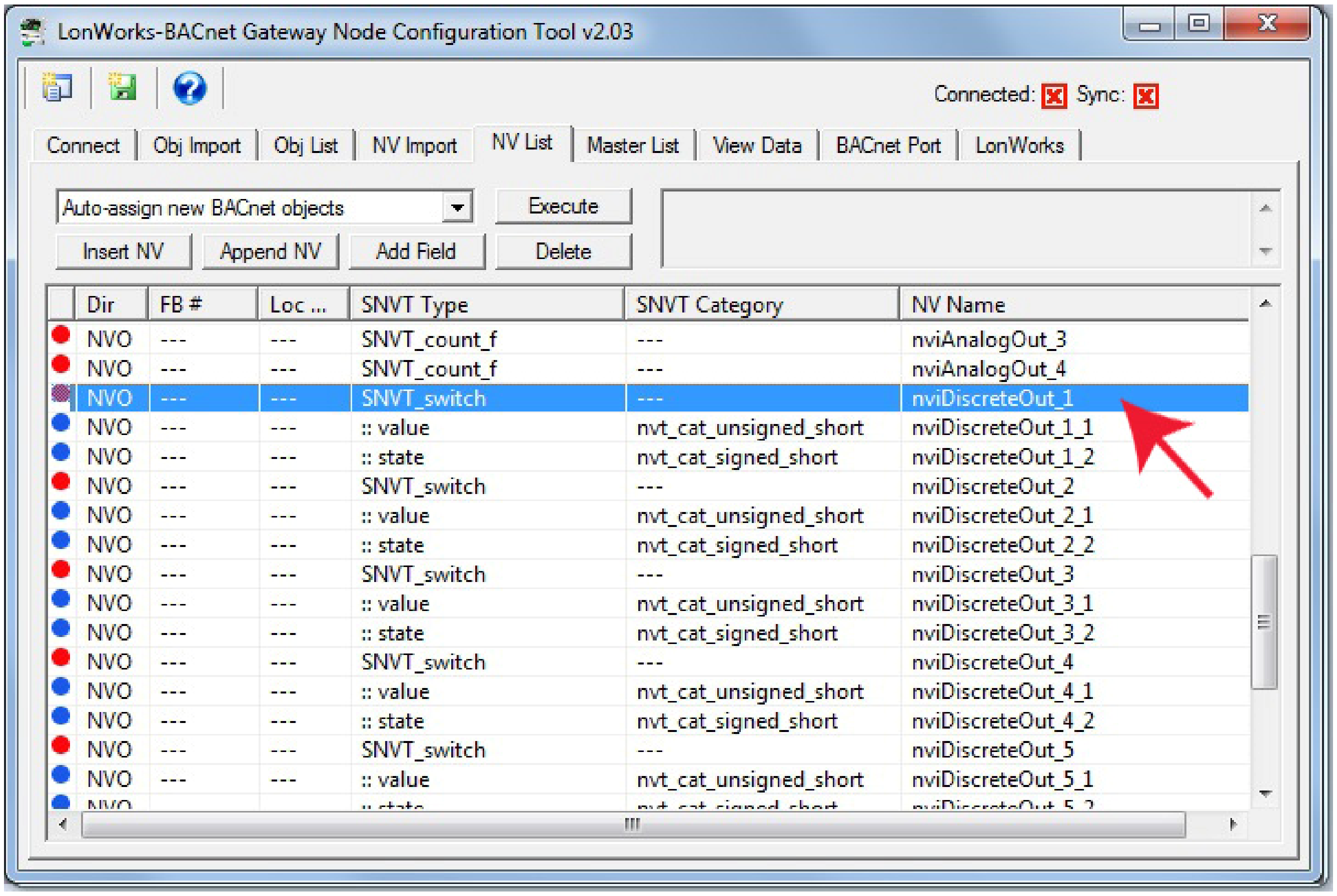


	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVI	---	---	SNVT_alarm2	---	nvoAlarm2
●	NVI	---	---	:: alarm_type	nvt_cat_enum	nvoAlarm2_1
●	NVI	---	---	:: priority_level	nvt_cat_enum	nvoAlarm2_2
●	NVI	---	---	:: alarm_time	nvt_cat_signed_quad	nvoAlarm2_3
●	NVI	---	---	:: milliseconds	nvt_cat_signed_long	nvoAlarm2_4
●	NVI	---	---	:: sequence_number	nvt_cat_unsigned_short	nvoAlarm2_5
●	NVI	---	---	:: description	nvt_cat_unsigned_char	nvoAlarm2_6
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_1
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_2
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_3
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_4
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_5
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_6
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_7
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_8
●	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_9

If you wish to make changes to the NV List, this is the point where you should do so. Do not proceed to assign function blocks (FB #'s) or data objects until the NV List is finalized.

You have a few options here, and these are described in more detail below. You may add network variables, delete them, or change what type they are. If they are structured, they will be automatically expanded into a list of all of their fields. If you add a structured NV which is not a standard LonMark type, you will need to add fields manually to build up the structure.

Some network variable types provide for special conversions. SNVT_switch is one such NV. To modify the NV definition, double click on the respective line in the NV List.


LonWorks-BACnet Gateway Node Configuration Tool v2.03

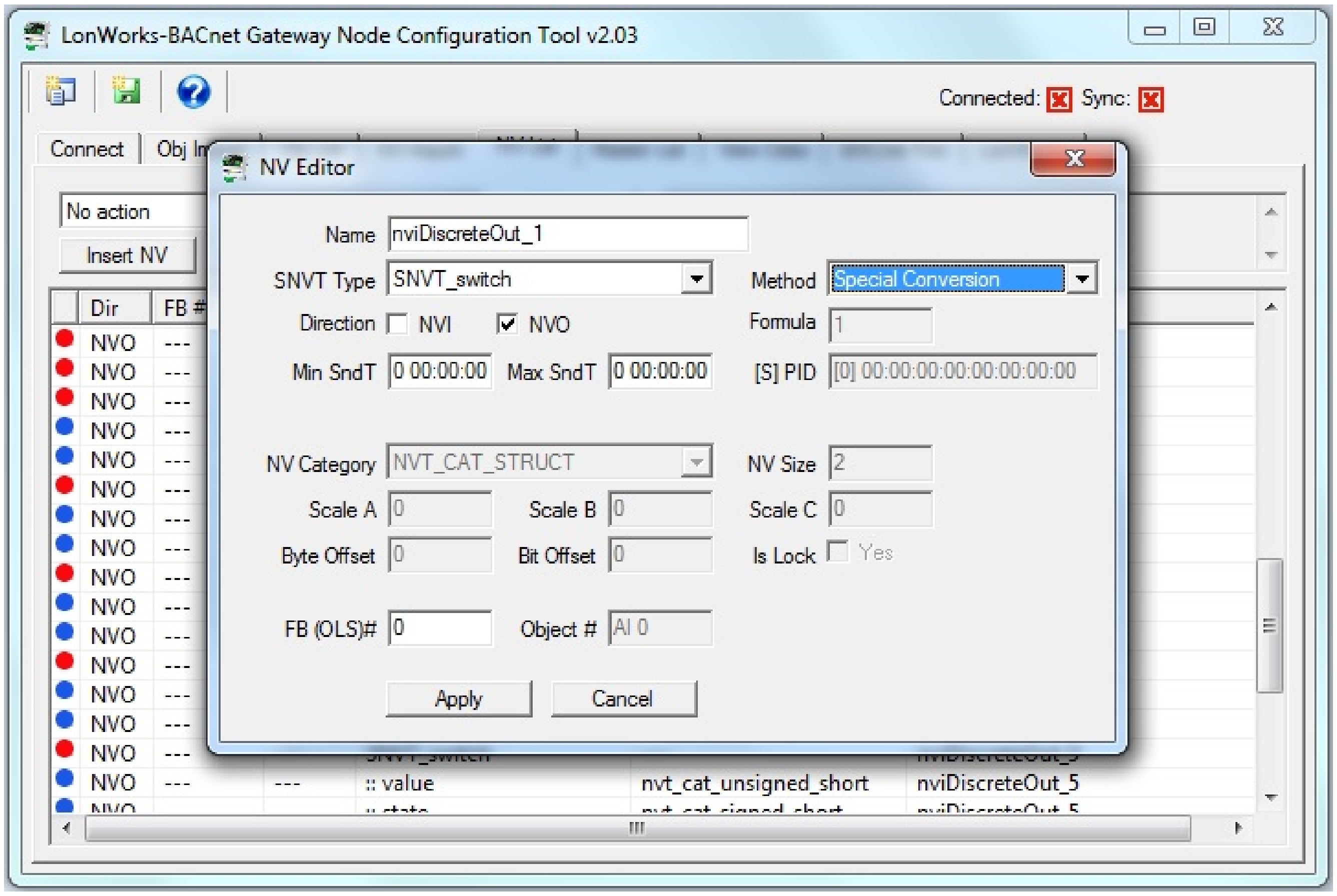
Connected: Sync:

Connect | Obj Import | Obj List | NV Import | NV List | Master List | View Data | BACnet Port | LonWorks

Auto-assign new BACnet objects | Execute | Insert NV | Append NV | Add Field | Delete

	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVO	---	---	SNVT_count_f	---	nviAnalogOut_3
●	NVO	---	---	SNVT_count_f	---	nviAnalogOut_4
▨	NVO	---	---	SNVT_switch	---	nviDiscreteOut_1
●	NVO	---	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_1_1
●	NVO	---	---	:: state	nvt_cat_signed_short	nviDiscreteOut_1_2
●	NVO	---	---	SNVT_switch	---	nviDiscreteOut_2
●	NVO	---	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_2_1
●	NVO	---	---	:: state	nvt_cat_signed_short	nviDiscreteOut_2_2
●	NVO	---	---	SNVT_switch	---	nviDiscreteOut_3
●	NVO	---	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_3_1
●	NVO	---	---	:: state	nvt_cat_signed_short	nviDiscreteOut_3_2
●	NVO	---	---	SNVT_switch	---	nviDiscreteOut_4
●	NVO	---	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_4_1
●	NVO	---	---	:: state	nvt_cat_signed_short	nviDiscreteOut_4_2
●	NVO	---	---	SNVT_switch	---	nviDiscreteOut_5
●	NVO	---	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_5_1
●	NVO			:: state	nvt_cat_signed_short	nviDiscreteOut_5_2

Upon double clicking a network variable in the NV List, the NV Editor dialog will appear. The SNVT_switch example is illustrated here. Your selection of whether to use the special conversion will decide whether SNVT_switch maps to one or two BACnet objects, and therefore whether SNVT_switch maps to one or two data objects. For this reason, it is important that you make all of your NV related selections and configurations BEFORE assigning function blocks and data objects.


LonWorks-BACnet Gateway Node Configuration Tool v2.03

Connected: ☒ Sync: ☒

NV Editor

Name: nviDiscreteOut_1

SNVT Type: SNVT_switch

Method: Special Conversion

Direction: ☐ NVI ☑ NVO

Formula: 1

Min SndT: 0 00:00:00

Max SndT: 0 00:00:00

[S] PID: [0] 00:00:00:00:00:00:00:00

NV Category: NVT_CAT_STRUCT

NV Size: 2

Scale A: 0

Scale B: 0

Scale C: 0

Byte Offset: 0

Bit Offset: 0

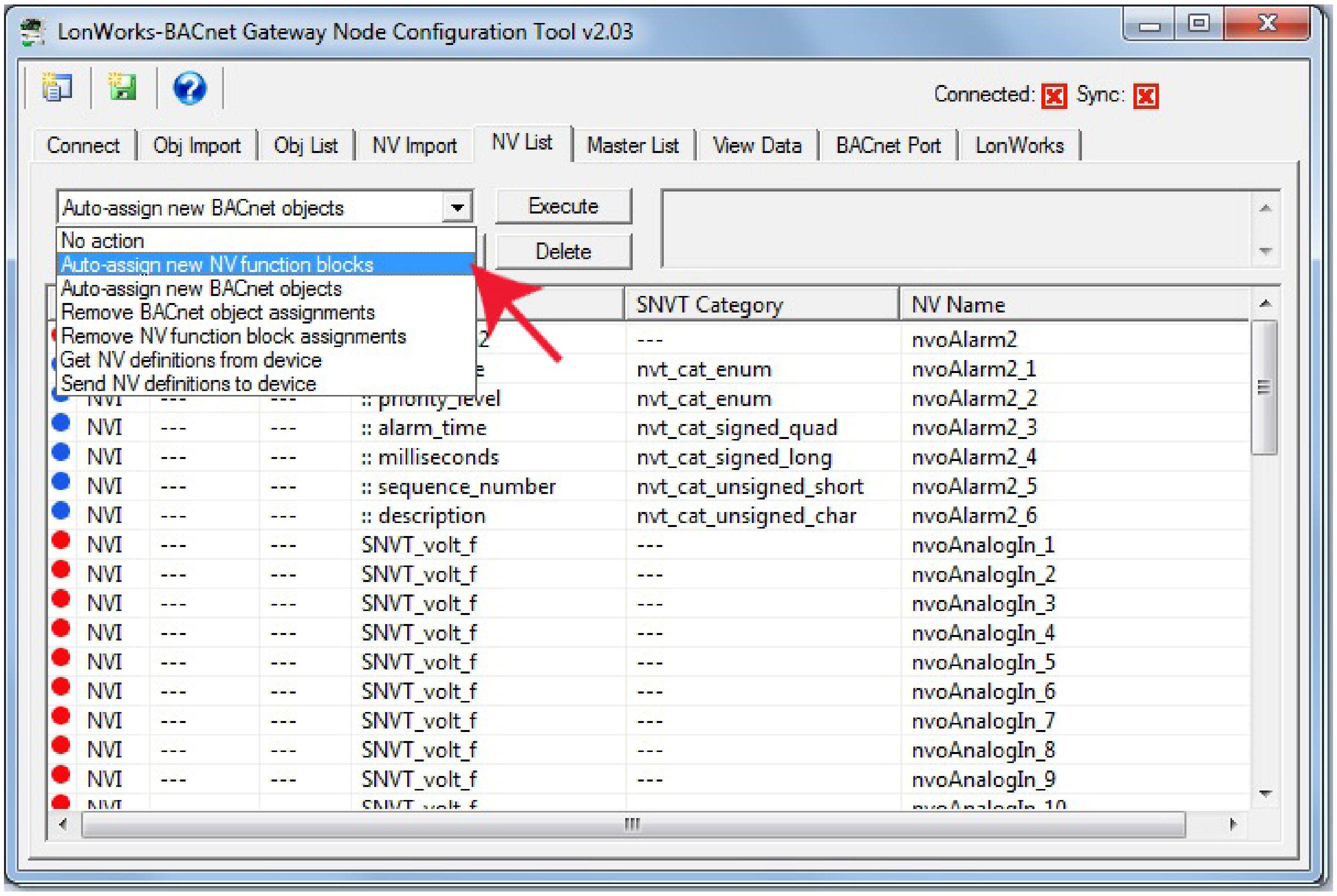
Is Lock: ☐ Yes

FB (OLS)#: 0

Object #: AI 0

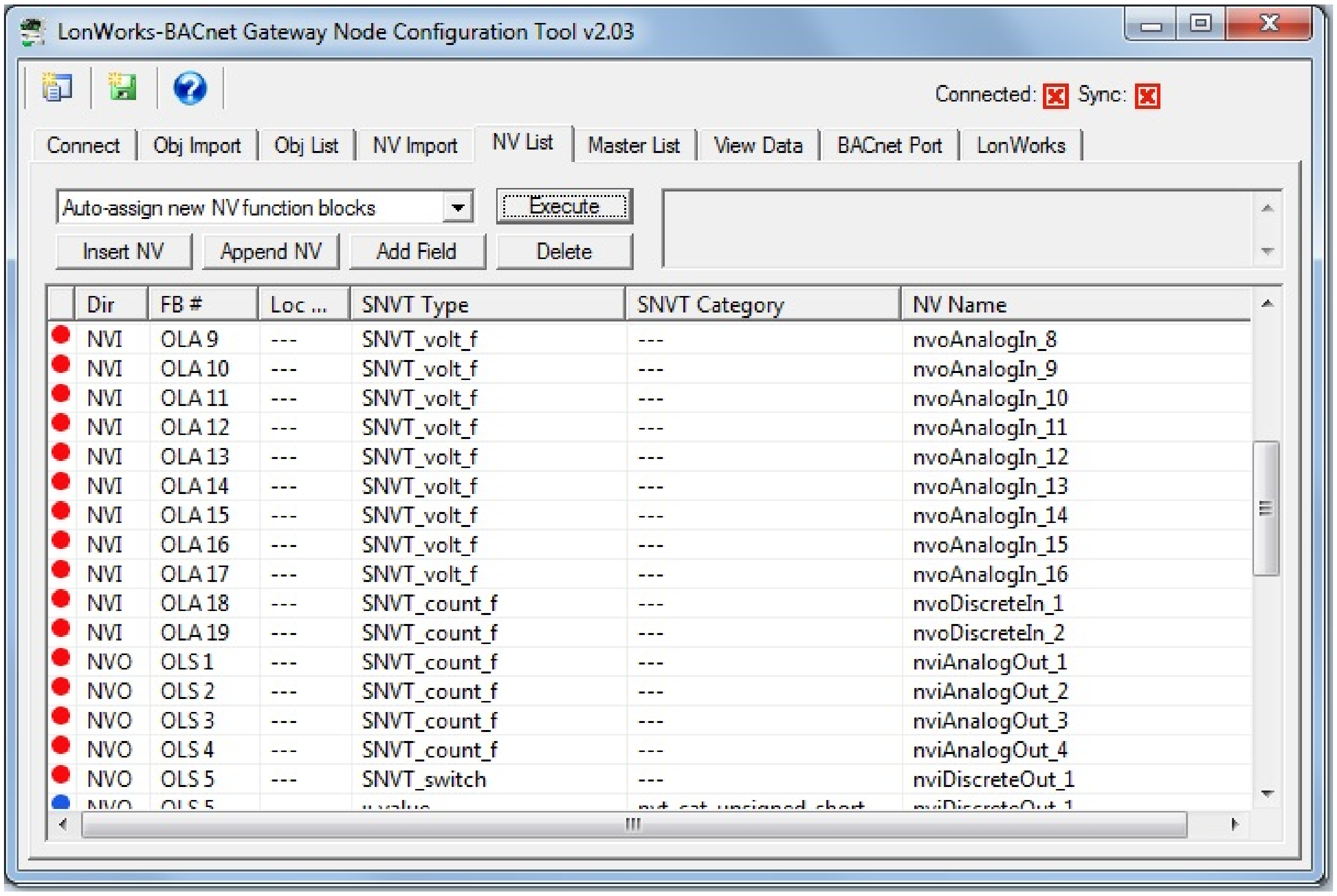
Dir	FB #				
● NVO	---				
● NVO	---				
● NVO	---				
● NVO	---				
● NVO	---				
● NVO	---				
● NVO	---				
● NVO	---				
● NVO	---				
● NVO	---				
● NVO	---				
● NVO	---				
● NVO	---				
● NVO	---				
● NVO	---				
● NVO	---	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_5

Once you have made all of the desired modifications to network variables in the NV List, proceed to assign function blocks. Select "Auto-assign new NV function blocks" and click Execute.


LonWorks-BACnet Gateway Node Configuration Tool v2.03
Connected:
Sync:
Connect
Obj Import
Obj List
NV Import
NV List
Master List
View Data
BACnet Port
LonWorks
Auto-assign new BACnet objects
Execute
Delete
No action
Auto-assign new NV function blocks
Auto-assign new BACnet objects
Remove BACnet object assignments
Remove NV function block assignments
Get NV definitions from device
Send NV definitions to device

					SNVT Category	NV Name
				2	---	nvoAlarm2
				e	nvt_cat_enum	nvoAlarm2_1
🔵	NVI	---	---	:: priority_level	nvt_cat_enum	nvoAlarm2_2
🔵	NVI	---	---	:: alarm_time	nvt_cat_signed_quad	nvoAlarm2_3
🔵	NVI	---	---	:: milliseconds	nvt_cat_signed_long	nvoAlarm2_4
🔵	NVI	---	---	:: sequence_number	nvt_cat_unsigned_short	nvoAlarm2_5
🔵	NVI	---	---	:: description	nvt_cat_unsigned_char	nvoAlarm2_6
🔴	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_1
🔴	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_2
🔴	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_3
🔴	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_4
🔴	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_5
🔴	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_6
🔴	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_7
🔴	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_8
🔴	NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_9
🔴	NVI			SNVT_volt_f		nvoAnalogIn_10

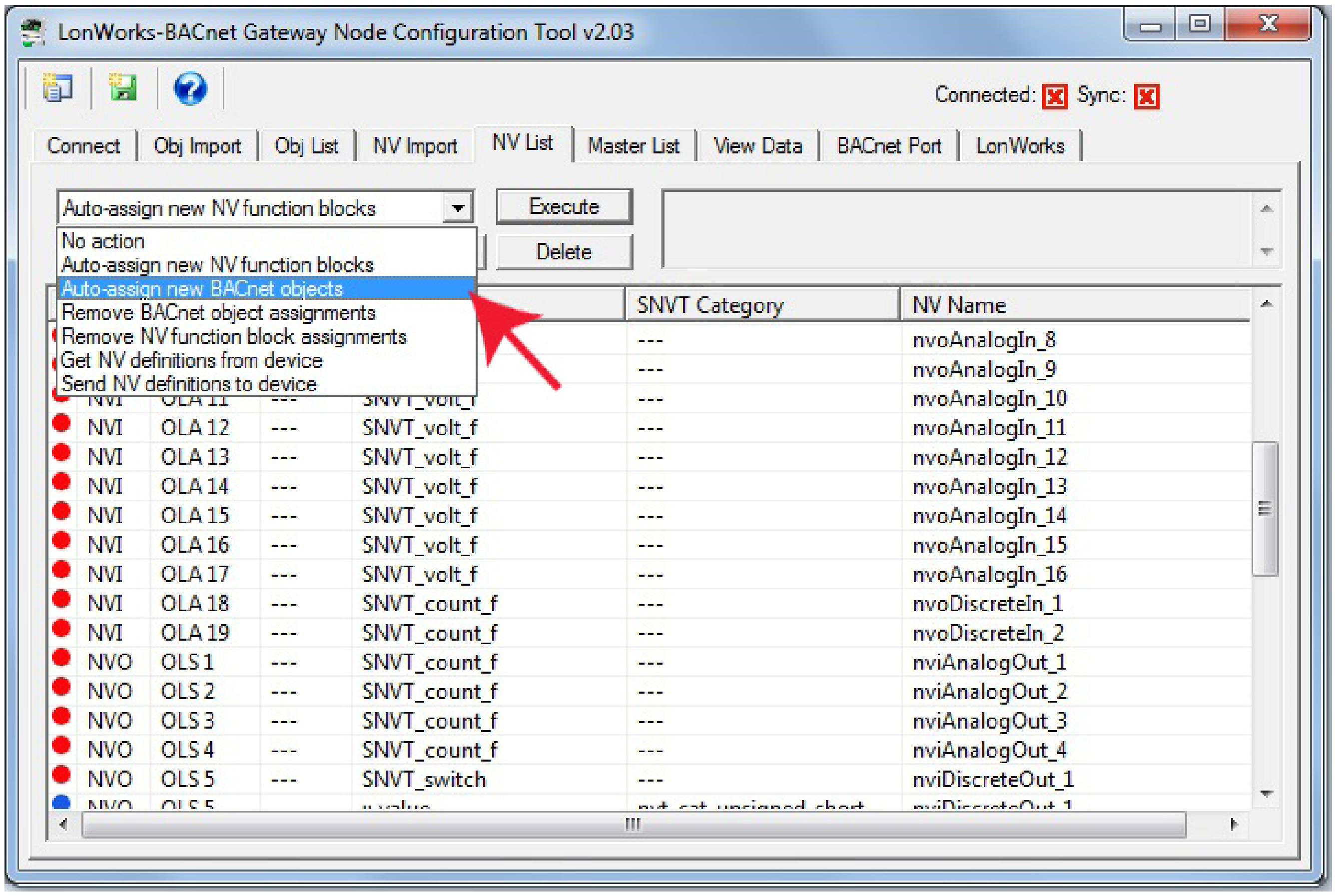
Upon execution of the function block assignments, the FB# column will be populated.


LonWorks-BACnet Gateway Node Configuration Tool v2.03

Connected: ☒ Sync: ☒

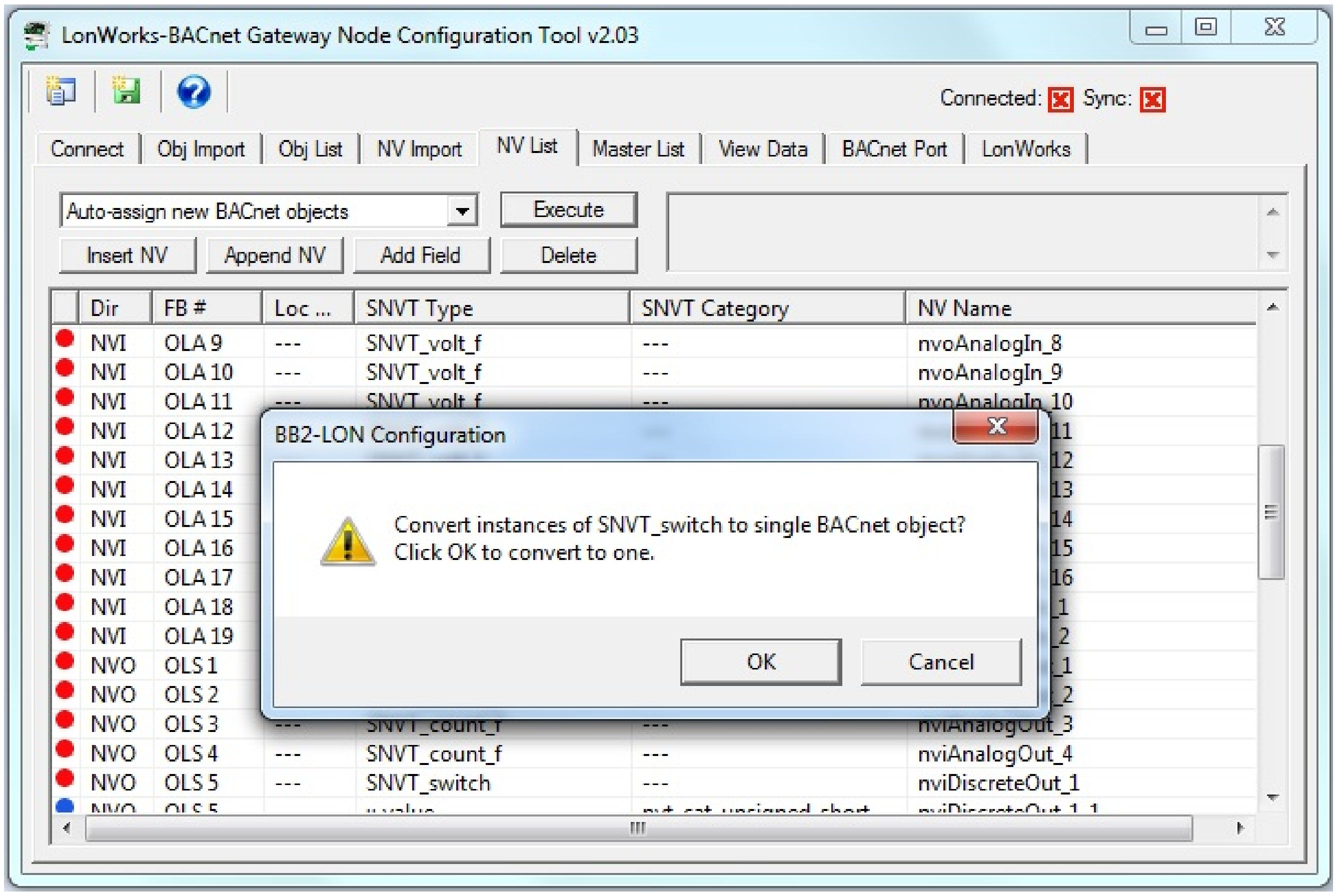
	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVI	OLA 9	---	SNVT_volt_f	---	nvoAnalogIn_8
●	NVI	OLA 10	---	SNVT_volt_f	---	nvoAnalogIn_9
●	NVI	OLA 11	---	SNVT_volt_f	---	nvoAnalogIn_10
●	NVI	OLA 12	---	SNVT_volt_f	---	nvoAnalogIn_11
●	NVI	OLA 13	---	SNVT_volt_f	---	nvoAnalogIn_12
●	NVI	OLA 14	---	SNVT_volt_f	---	nvoAnalogIn_13
●	NVI	OLA 15	---	SNVT_volt_f	---	nvoAnalogIn_14
●	NVI	OLA 16	---	SNVT_volt_f	---	nvoAnalogIn_15
●	NVI	OLA 17	---	SNVT_volt_f	---	nvoAnalogIn_16
●	NVI	OLA 18	---	SNVT_count_f	---	nvoDiscreteIn_1
●	NVI	OLA 19	---	SNVT_count_f	---	nvoDiscreteIn_2
●	NVO	OLS 1	---	SNVT_count_f	---	nviAnalogOut_1
●	NVO	OLS 2	---	SNVT_count_f	---	nviAnalogOut_2
●	NVO	OLS 3	---	SNVT_count_f	---	nviAnalogOut_3
●	NVO	OLS 4	---	SNVT_count_f	---	nviAnalogOut_4
●	NVO	OLS 5	---	SNVT_switch	---	nviDiscreteOut_1
●	NVO	OLS 5		u.value	nvt_cat_unsigned_short	nviDiscreteOut_1

Next, select and execute "Auto-assign new BACnet objects".


LonWorks-BACnet Gateway Node Configuration Tool v2.03
Connected: Sync:
Connect
Obj Import
Obj List
NV Import
NV List
Master List
View Data
BACnet Port
LonWorks
Auto-assign new NV function blocks
Execute
Delete
No action
Auto-assign new NV function blocks
Auto-assign new BACnet objects
Remove BACnet object assignments
Remove NV function block assignments
Get NV definitions from device
Send NV definitions to device

					SNVT Category	NV Name
●					---	nvoAnalogIn_8
●					---	nvoAnalogIn_9
●	NVI	OLA 11	---	SNVT_volt_f	---	nvoAnalogIn_10
●	NVI	OLA 12	---	SNVT_volt_f	---	nvoAnalogIn_11
●	NVI	OLA 13	---	SNVT_volt_f	---	nvoAnalogIn_12
●	NVI	OLA 14	---	SNVT_volt_f	---	nvoAnalogIn_13
●	NVI	OLA 15	---	SNVT_volt_f	---	nvoAnalogIn_14
●	NVI	OLA 16	---	SNVT_volt_f	---	nvoAnalogIn_15
●	NVI	OLA 17	---	SNVT_volt_f	---	nvoAnalogIn_16
●	NVI	OLA 18	---	SNVT_count_f	---	nvoDiscreteIn_1
●	NVI	OLA 19	---	SNVT_count_f	---	nvoDiscreteIn_2
●	NVO	OLS 1	---	SNVT_count_f	---	nviAnalogOut_1
●	NVO	OLS 2	---	SNVT_count_f	---	nviAnalogOut_2
●	NVO	OLS 3	---	SNVT_count_f	---	nviAnalogOut_3
●	NVO	OLS 4	---	SNVT_count_f	---	nviAnalogOut_4
●	NVO	OLS 5	---	SNVT_switch	---	nviDiscreteOut_1
●	NVO	OLS 5		u-value	nvt_cat_unsigned_short	nviDiscreteOut_1

Upon clicking Execute, you will be given the option of automatically converting all instances of SNVT_switch to single objects. This is a short cut alternative to the manual editing of SNVT_switch variables noted above.

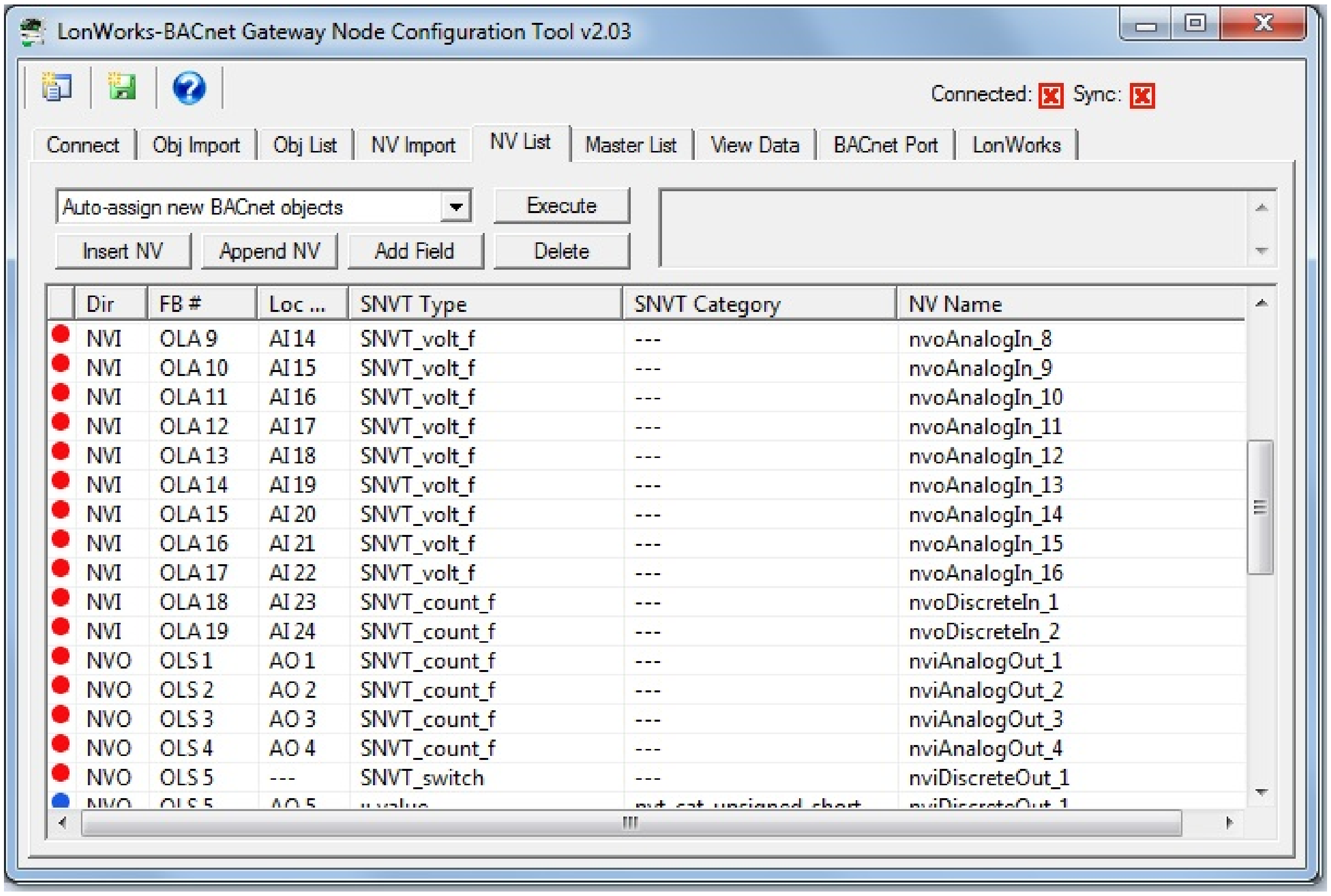


	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
🔴	NVI	OLA 9	---	SNVT_volt_f	---	nvoAnalogIn_8
🔴	NVI	OLA 10	---	SNVT_volt_f	---	nvoAnalogIn_9
🔴	NVI	OLA 11	---	SNVT_volt_f	---	nvoAnalogIn_10
🔴	NVI	OLA 12				11
🔴	NVI	OLA 13				12
🔴	NVI	OLA 14				13
🔴	NVI	OLA 15				14
🔴	NVI	OLA 16				15
🔴	NVI	OLA 17				16
🔴	NVI	OLA 18				1
🔴	NVI	OLA 19				2
🔴	NVO	OLS 1				1
🔴	NVO	OLS 2				2
🔴	NVO	OLS 3	---	SNVT_count_f	---	nviAnalogOut_3
🔴	NVO	OLS 4	---	SNVT_count_f	---	nviAnalogOut_4
🔴	NVO	OLS 5	---	SNVT_switch	---	nviDiscreteOut_1
🔵	NVO	OLS 5		u value	nvt_cat_unsigned_short	nviDiscreteOut_1_1

BB2-LON Configuration

Convert instances of SNVT_switch to single BACnet object?
Click OK to convert to one.

After BACnet object assignment, the Loc... (Local Object) column will be populated.



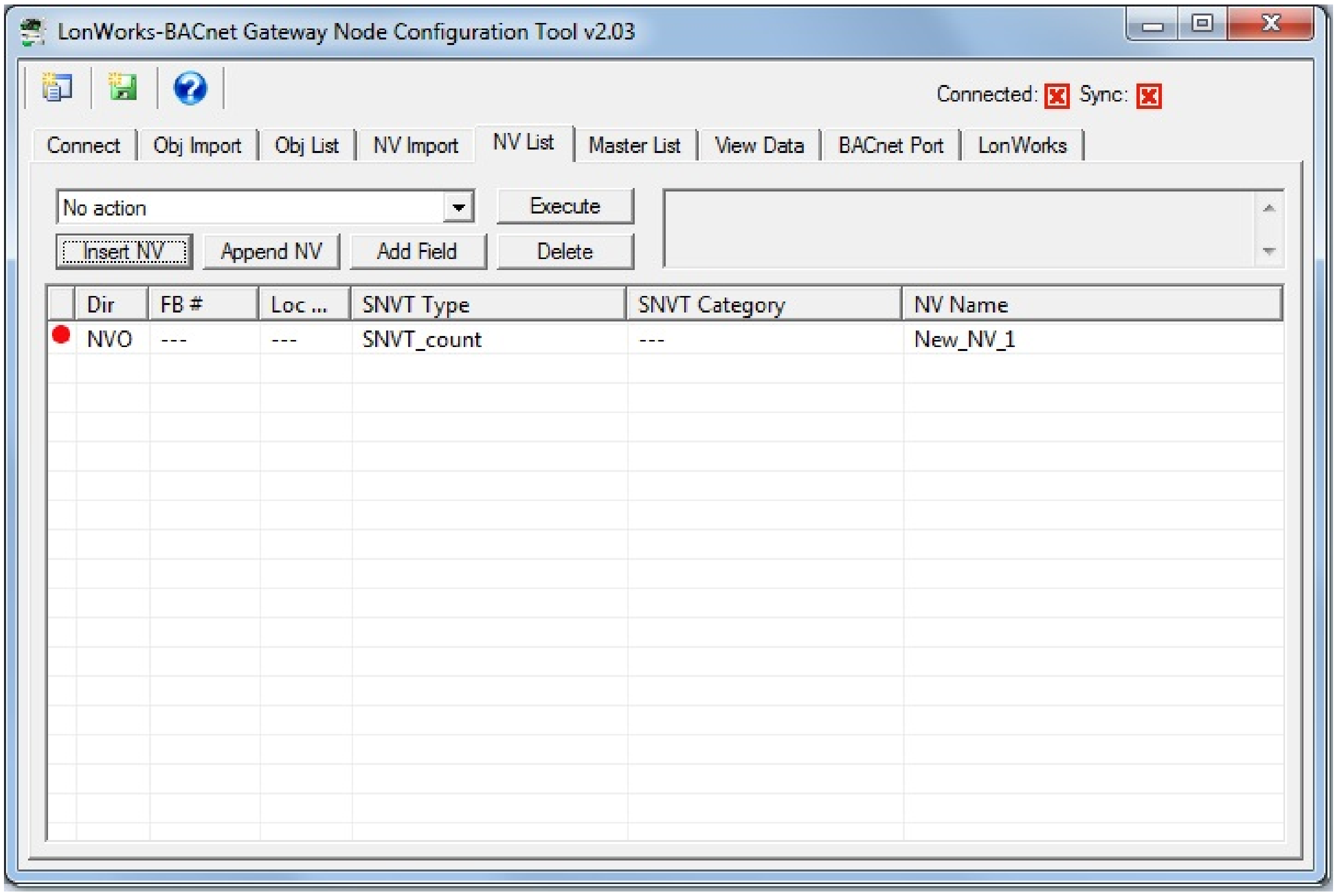
	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVI	OLA 9	AI 14	SNVT_volt_f	---	nvoAnalogIn_8
●	NVI	OLA 10	AI 15	SNVT_volt_f	---	nvoAnalogIn_9
●	NVI	OLA 11	AI 16	SNVT_volt_f	---	nvoAnalogIn_10
●	NVI	OLA 12	AI 17	SNVT_volt_f	---	nvoAnalogIn_11
●	NVI	OLA 13	AI 18	SNVT_volt_f	---	nvoAnalogIn_12
●	NVI	OLA 14	AI 19	SNVT_volt_f	---	nvoAnalogIn_13
●	NVI	OLA 15	AI 20	SNVT_volt_f	---	nvoAnalogIn_14
●	NVI	OLA 16	AI 21	SNVT_volt_f	---	nvoAnalogIn_15
●	NVI	OLA 17	AI 22	SNVT_volt_f	---	nvoAnalogIn_16
●	NVI	OLA 18	AI 23	SNVT_count_f	---	nvoDiscreteIn_1
●	NVI	OLA 19	AI 24	SNVT_count_f	---	nvoDiscreteIn_2
●	NVO	OLS 1	AO 1	SNVT_count_f	---	nviAnalogOut_1
●	NVO	OLS 2	AO 2	SNVT_count_f	---	nviAnalogOut_2
●	NVO	OLS 3	AO 3	SNVT_count_f	---	nviAnalogOut_3
●	NVO	OLS 4	AO 4	SNVT_count_f	---	nviAnalogOut_4
●	NVO	OLS 5	---	SNVT_switch	---	nviDiscreteOut_1
●	NVO	OLS 5	AO 5	.value	nvt_cat_unsigned_short	nviDiscreteOut_1

8.3 Building Configuration Manually

It is not required that you start with either a CSV file or an XIF file. You can use the configuration tool to start with a blank slate and build your configuration from scratch.

It is assumed that you have some familiarity with BACnet, and also an understanding of LonWorks network variables. You will need to obtain a copy of the documentation of SNVT types from LonMark (www.lonmark.org) in order to have any success in creating the LonWorks side of the gateway configuration.

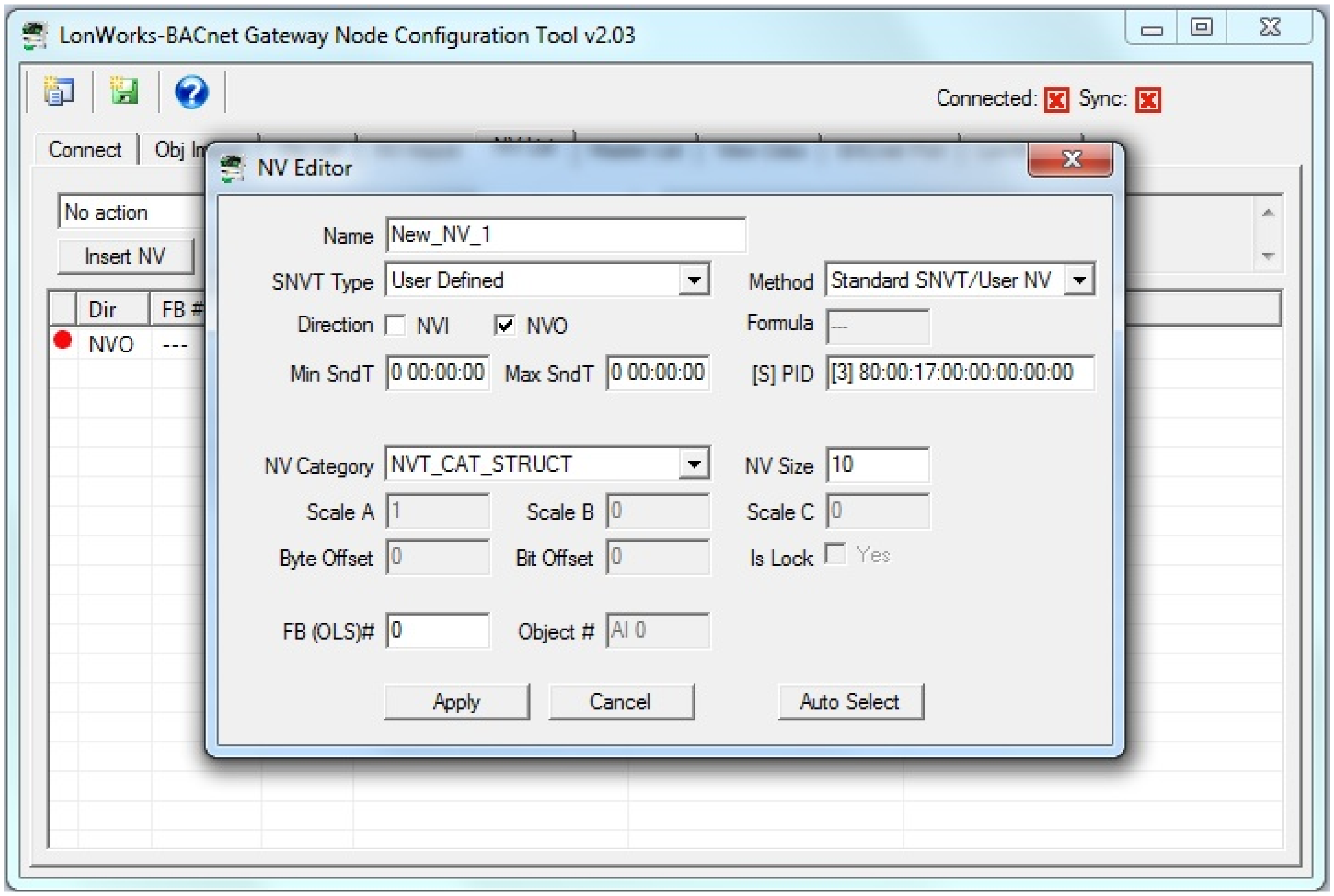
For each network variable you wish to add, click Append NV to add one at the end of the list, or Insert NV to add one immediately before the selected line on the list (if any).



	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVO	---	---	SNVT_count	---	New_NV_1

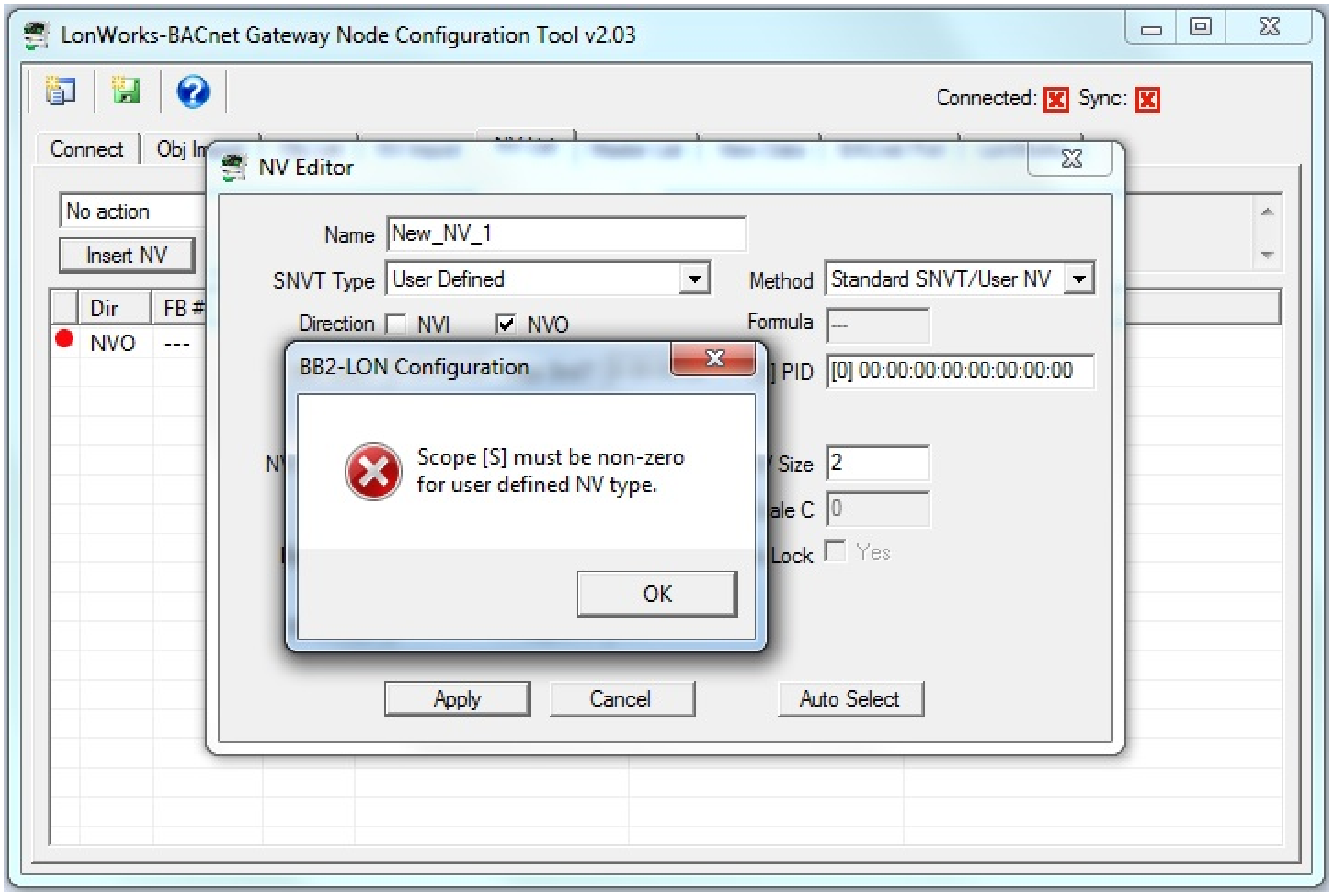
The newly inserted NV will default to SNVT_count. If you wish to change it, double click on the line to be changed, and the NV Editor dialog will appear. In the example that follows, we will complicate things to the maximum by creating a user defined structured network variable.

When creating a structure, the NV needs to be set to NV Category NVT_CAT_STRUCT and the NV Size needs to be set to the number of bytes that make up this variable in the LonWorks device. You also need to change the Scope [S] and PID (program ID) to something unique that defines the manufacturer of the device you are conforming to.



Name: New_NV_1
SNVT Type: User Defined
Method: Standard SNVT/User NV
Direction: NVI [✓ NVO]
Formula: ---
Min SndT: 0 00:00:00
Max SndT: 0 00:00:00
[S] PID: [3] 80:00:17:00:00:00:00:00
NV Category: NVT_CAT_STRUCT
NV Size: 10
Scale A: 1
Scale B: 0
Scale C: 0
Byte Offset: 0
Bit Offset: 0
Is Lock: Yes
FB (OLS)#: 0
Object #: AI 0

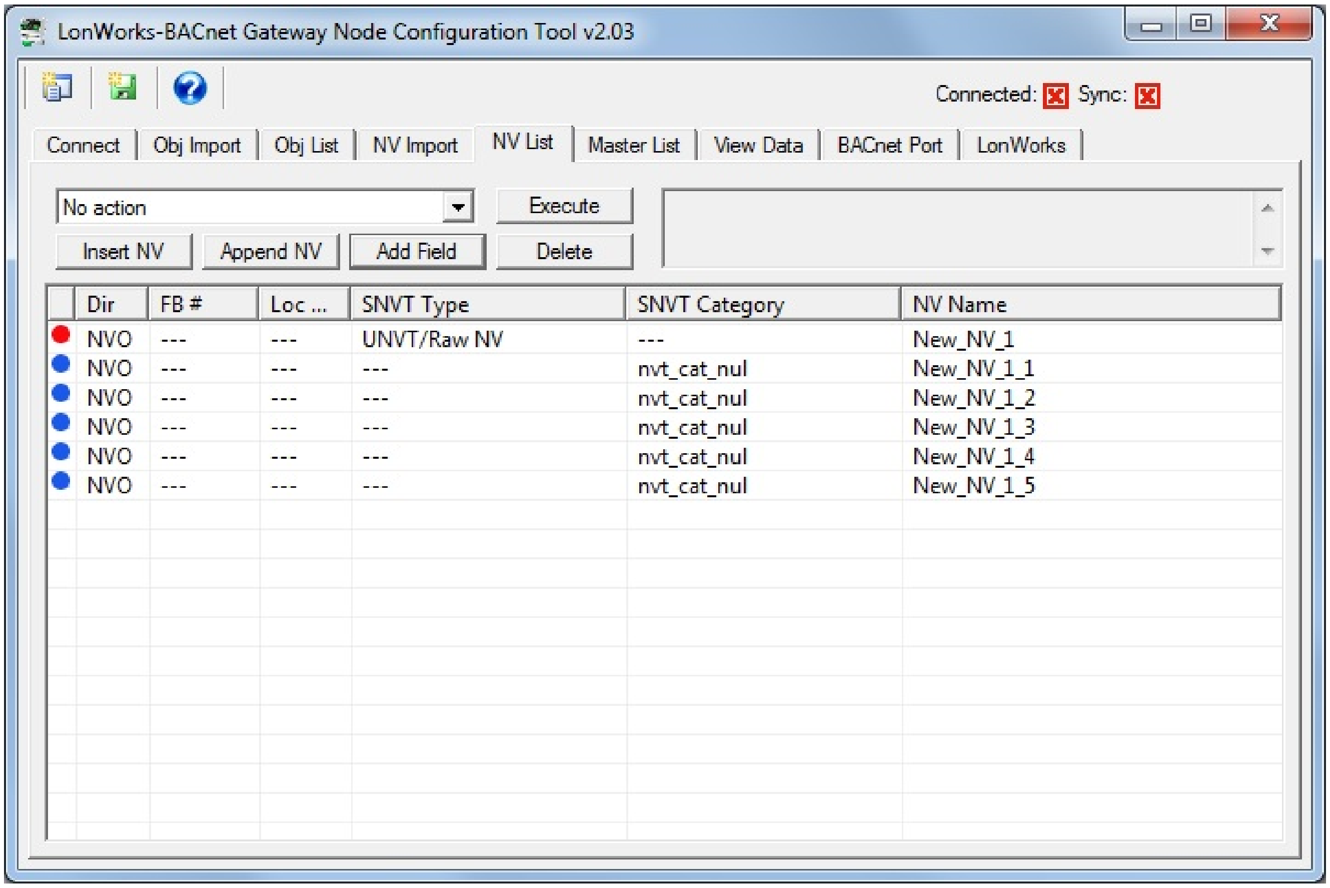
If you do not change the scope, most commonly to 3 for 'manufacturer defined', you will get an error message and not be allowed to proceed. If the scope remains at zero, then the gateway (and all other LonWorks devices on the network) will attempt to interpret your NV as a standard LonMark type and data conversion results will be wrong.



Name: New_NV_1
SNVT Type: User Defined
Method: Standard SNVT/User NV
Direction: NVI ☐ NVO ☑
Formula: ---
PID: [0] 00:00:00:00:00:00:00:00
Size: 2
ale C: 0
Lock: ☐ Yes

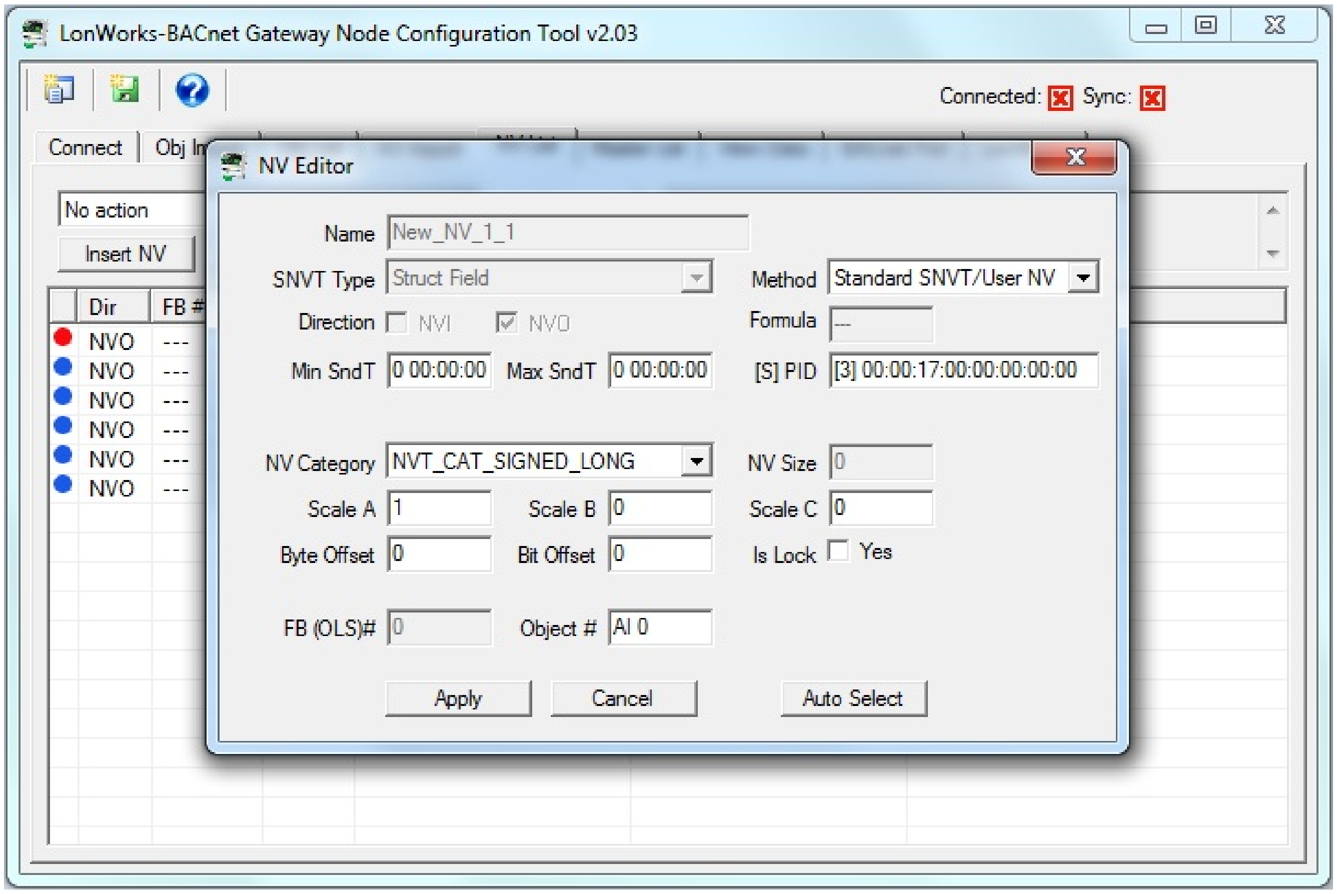
Scope [S] must be non-zero for user defined NV type.

After defining the NV as a structure, you next need to add fields to the structure. Do this by clicking Add Field.



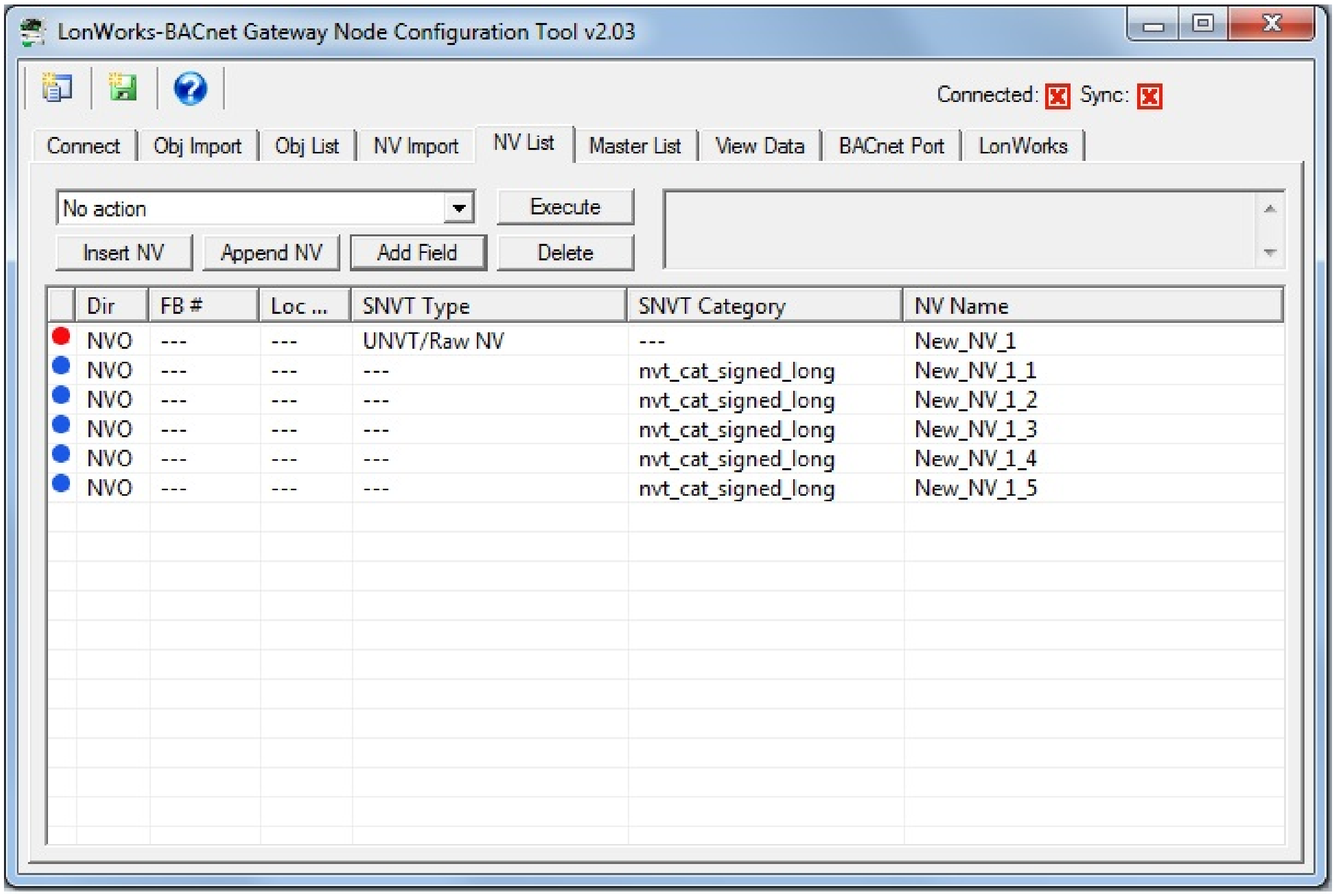
	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
🔴	NVO	---	---	UNVT/Raw NV	---	New_NV_1
🔵	NVO	---	---	---	nvt_cat_nul	New_NV_1_1
🔵	NVO	---	---	---	nvt_cat_nul	New_NV_1_2
🔵	NVO	---	---	---	nvt_cat_nul	New_NV_1_3
🔵	NVO	---	---	---	nvt_cat_nul	New_NV_1_4
🔵	NVO	---	---	---	nvt_cat_nul	New_NV_1_5

Next, edit each field by double clicking on that line. Specify the data type for the field. It is also important to specify the Byte Offset in the structure. An offset of zero means this field occupies the first byte(s) of the structure. You also need to enter the scale values. These follow the LonMark definition of scale.



Name	New_NV_1_1				
SNVT Type	Struct Field			Method	Standard SNVT/User NV
Direction	☐ NVI	☑ NVO		Formula	---
Min SndT	0 00:00:00	Max SndT	0 00:00:00	[S] PID	[3] 00:00:17:00:00:00:00:00
NV Category	NVT_CAT_SIGNED_LONG			NV Size	0
Scale A	1	Scale B	0	Scale C	0
Byte Offset	0	Bit Offset	0	Is Lock	☐ Yes
FB (OLS)#	0	Object #	AI 0		

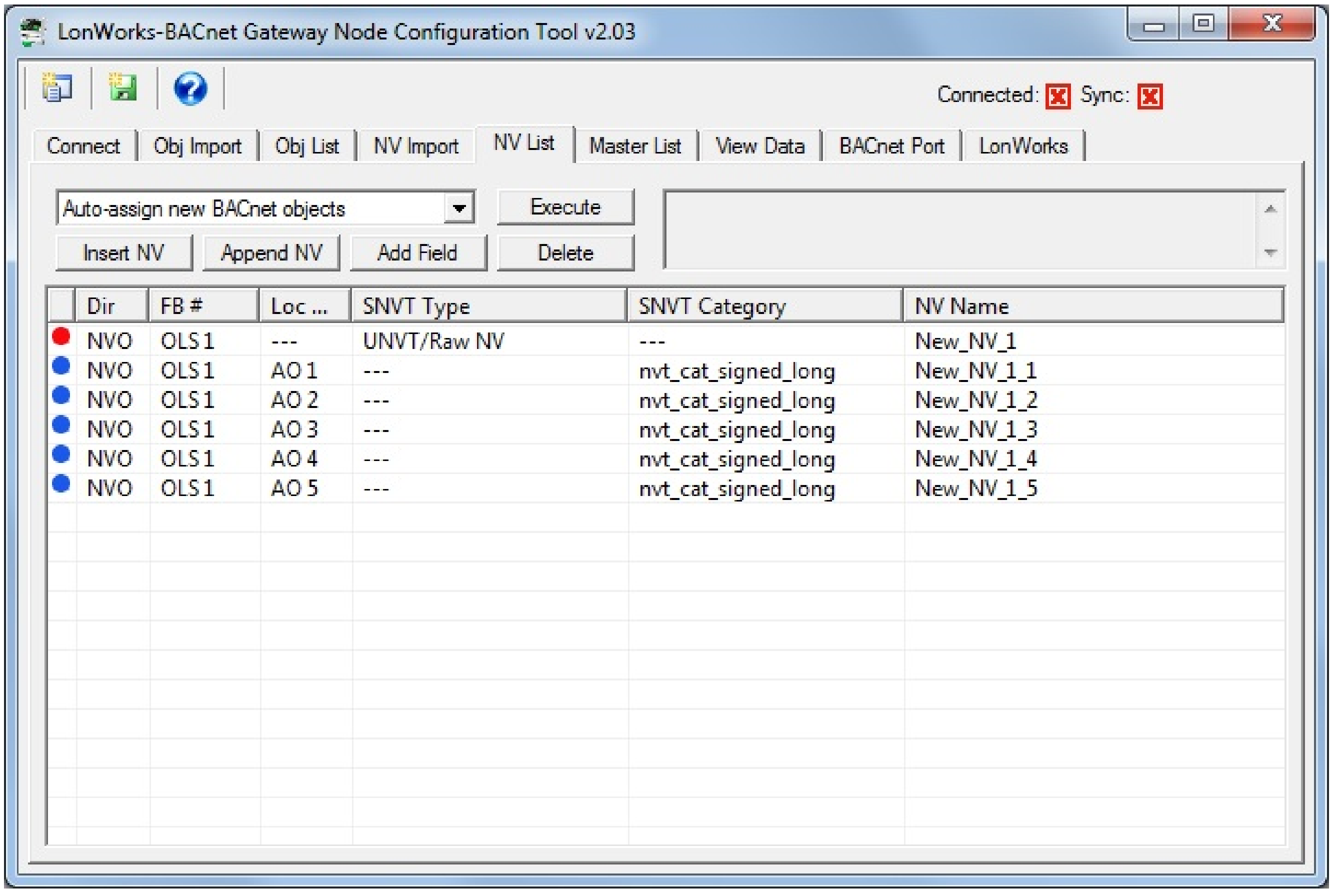
Following configuration of the NV and its fields, your NV List will appear as follows for this first single NV that will map to five BACnet objects.


LonWorks-BACnet Gateway Node Configuration Tool v2.03

Connected: ☒ Sync: ☒

	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
🔴	NVO	---	---	UNVT/Raw NV	---	New_NV_1
🔵	NVO	---	---	---	nvt_cat_signed_long	New_NV_1_1
🔵	NVO	---	---	---	nvt_cat_signed_long	New_NV_1_2
🔵	NVO	---	---	---	nvt_cat_signed_long	New_NV_1_3
🔵	NVO	---	---	---	nvt_cat_signed_long	New_NV_1_4
🔵	NVO	---	---	---	nvt_cat_signed_long	New_NV_1_5

If you have no further network variables to add, proceed with the auto-build of the rest of the configuration. First execute "Auto-assign new NV function blocks". Then execute "Auto-assign new BACnet objects". Upon completion of these steps, the NV List would appear as follows for this example.

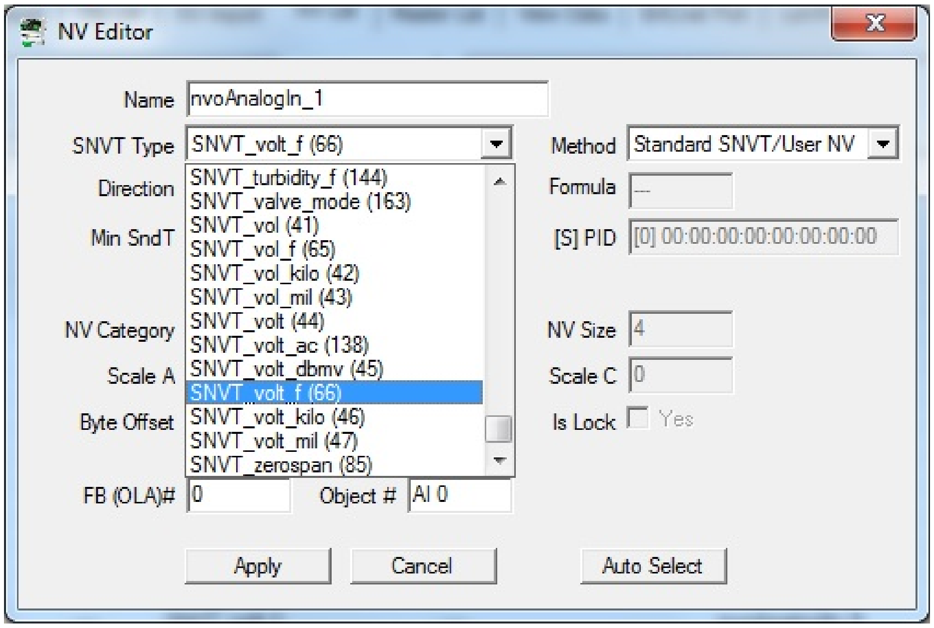


	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
🔴	NVO	OLS 1	---	UNVT/Raw NV	---	New_NV_1
🔵	NVO	OLS 1	AO 1	---	nvt_cat_signed_long	New_NV_1_1
🔵	NVO	OLS 1	AO 2	---	nvt_cat_signed_long	New_NV_1_2
🔵	NVO	OLS 1	AO 3	---	nvt_cat_signed_long	New_NV_1_3
🔵	NVO	OLS 1	AO 4	---	nvt_cat_signed_long	New_NV_1_4
🔵	NVO	OLS 1	AO 5	---	nvt_cat_signed_long	New_NV_1_5

8.4 Using the NV Editor

The layout of the NV Editor will be the same in all cases, but certain windows in the dialog will be set to 'read-only' in some cases depending on what is being configured.

The following example illustrates the most simple of the NV Editor operations, namely just selecting a different standard SNVT type from the list of SNVT Types. When a standard LonMark SNVT type is selected, most of the configuration parameters will be fixed by LonMark and set to read-only in the editor dialog. Click Apply to accept the changes, or Cancel to discard.


NV Editor

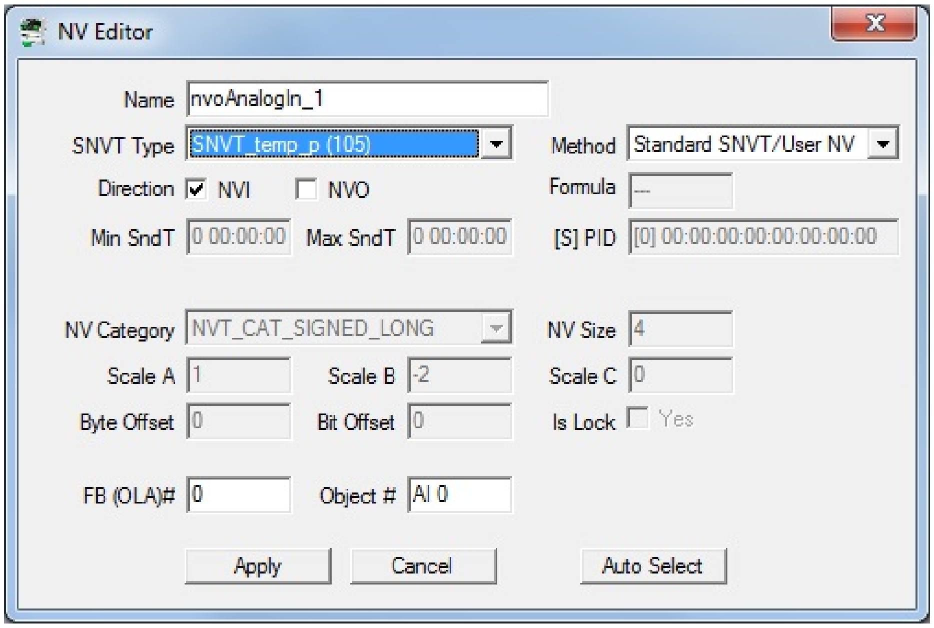
Name: nvoAnalogIn_1

SNVT Type: SNVT_volt_f (66)

SNVT_turbidity_f (144)
SNVT_valve_mode (163)
SNVT_vol (41)
SNVT_vol_f (65)
SNVT_vol_kilo (42)
SNVT_vol_mil (43)
SNVT_volt (44)
SNVT_volt_ac (138)
SNVT_volt_dbmv (45)
SNVT_volt_f (66)
SNVT_volt_kilo (46)
SNVT_volt_mil (47)
SNVT_zerospan (85)

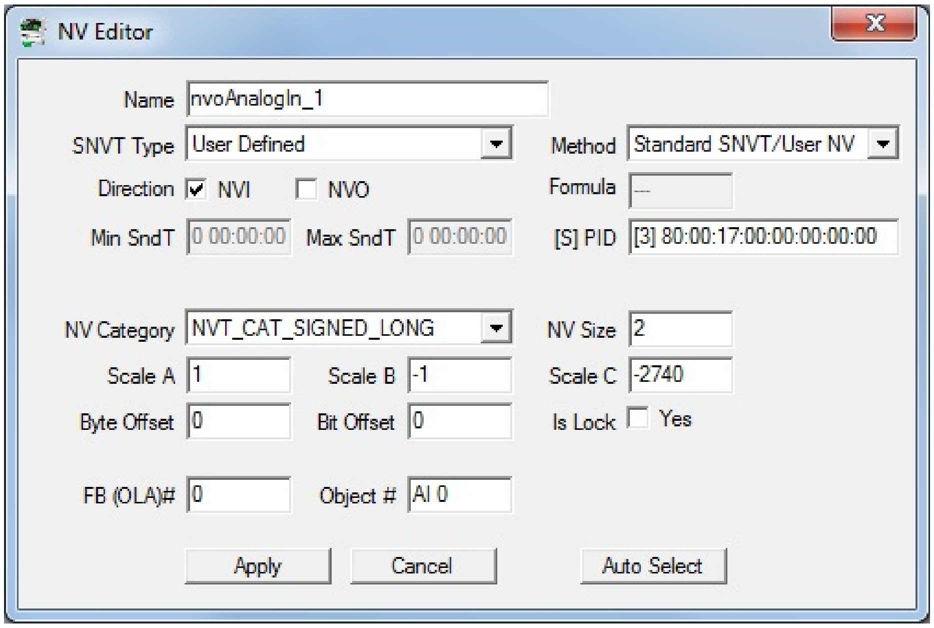
Method: Standard SNVT/User NV
Formula: ---
[S] PID: [0] 00:00:00:00:00:00:00:00
NV Size: 4
Scale C: 0
Is Lock: ☐ Yes
FB (OLA)#: 0
Object #: AI 0

The following is an example of a standard SNVT selection:


NV Editor

Name: nvoAnalogIn_1
SNVT Type: SNVT_temp_p (105)
Method: Standard SNVT/User NV
Direction: ☑ NVI ☐ NVO
Formula: ---
Min SndT: 0 00:00:00
Max SndT: 0 00:00:00
[S] PID: [0] 00:00:00:00:00:00:00:00
NV Category: NVT_CAT_SIGNED_LONG
NV Size: 4
Scale A: 1
Scale B: -2
Scale C: 0
Byte Offset: 0
Bit Offset: 0
Is Lock: ☐ Yes
FB (OLA)#: 0
Object #: AI 0

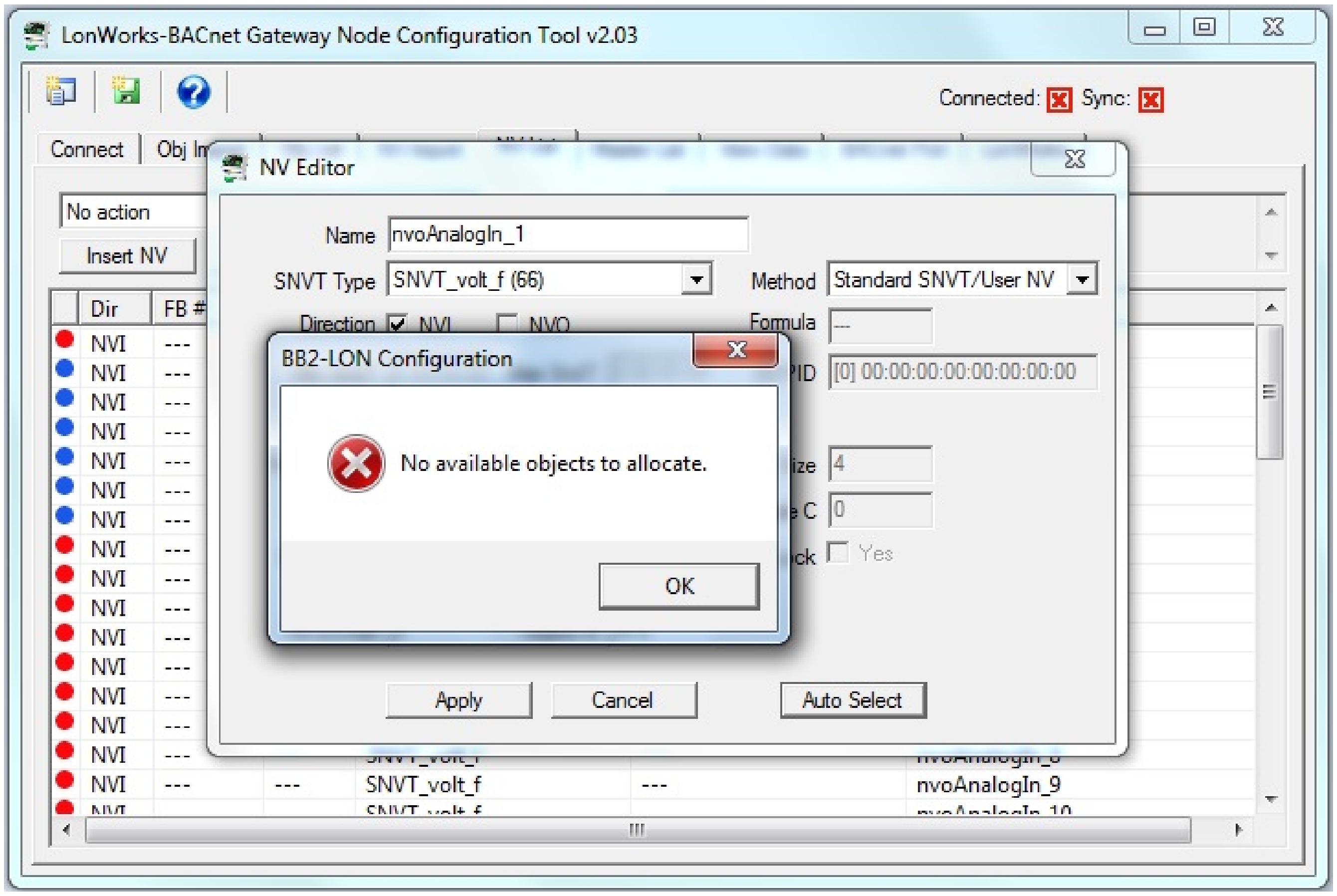
If you were to create a 'user defined' NV that provided the exact same results as SNVT_temp, the NV Editor dialog would appear as follows:


NV Editor

Name: nvoAnalogIn_1
SNVT Type: User Defined
Method: Standard SNVT/User NV
Direction: ☑ NVI ☐ NVO
Formula: ---
Min SndT: 0 00:00:00
Max SndT: 0 00:00:00
[S] PID: [3] 80:00:17:00:00:00:00:00
NV Category: NVT_CAT_SIGNED_LONG
NV Size: 2
Scale A: 1
Scale B: -1
Scale C: -2740
Byte Offset: 0
Bit Offset: 0
Is Lock: ☐ Yes
FB (OLA)#: 0
Object #: AI 0

You will note that the Function Block (FB#) and Object numbers have not yet been assigned in the above examples. To assign them, enter numbers if available numbers are known. To have the tool suggest numbers, click Auto Select.

If you click Auto Select, but there are not enough unallocated objects, you will see the following:


LonWorks-BACnet Gateway Node Configuration Tool v2.03

Connected: ☒ Sync: ☒

NV Editor

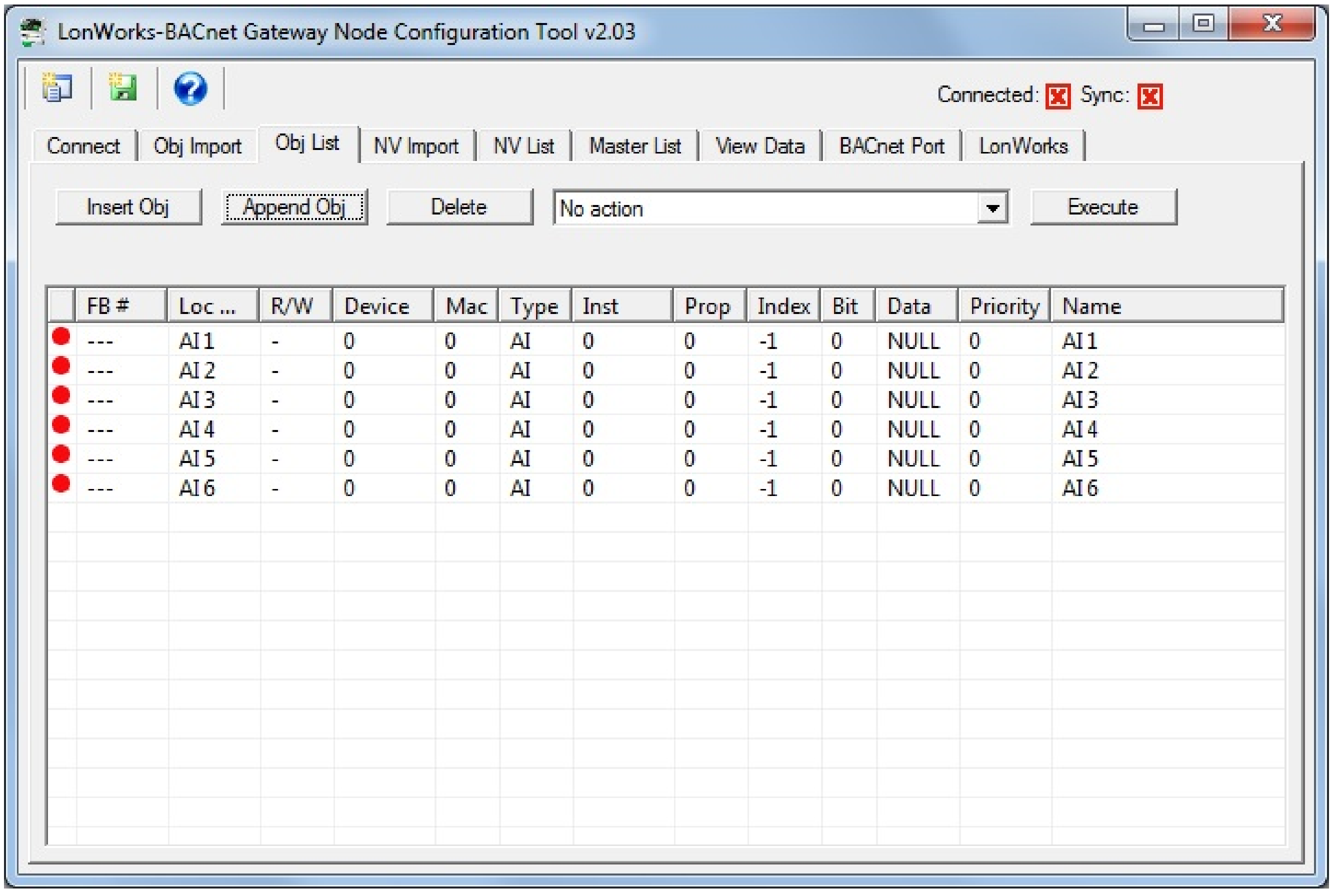
Name: nvoAnalogIn_1
SNVT Type: SNVT_volt_f (66)
Method: Standard SNVT/User NV
Direction: ☑ NVI ☐ NVO
Formula: ---
PID: [0] 00:00:00:00:00:00:00:00
Size: 4
C: 0
Lock: ☐ Yes

BB2-LON Configuration

No available objects to allocate.

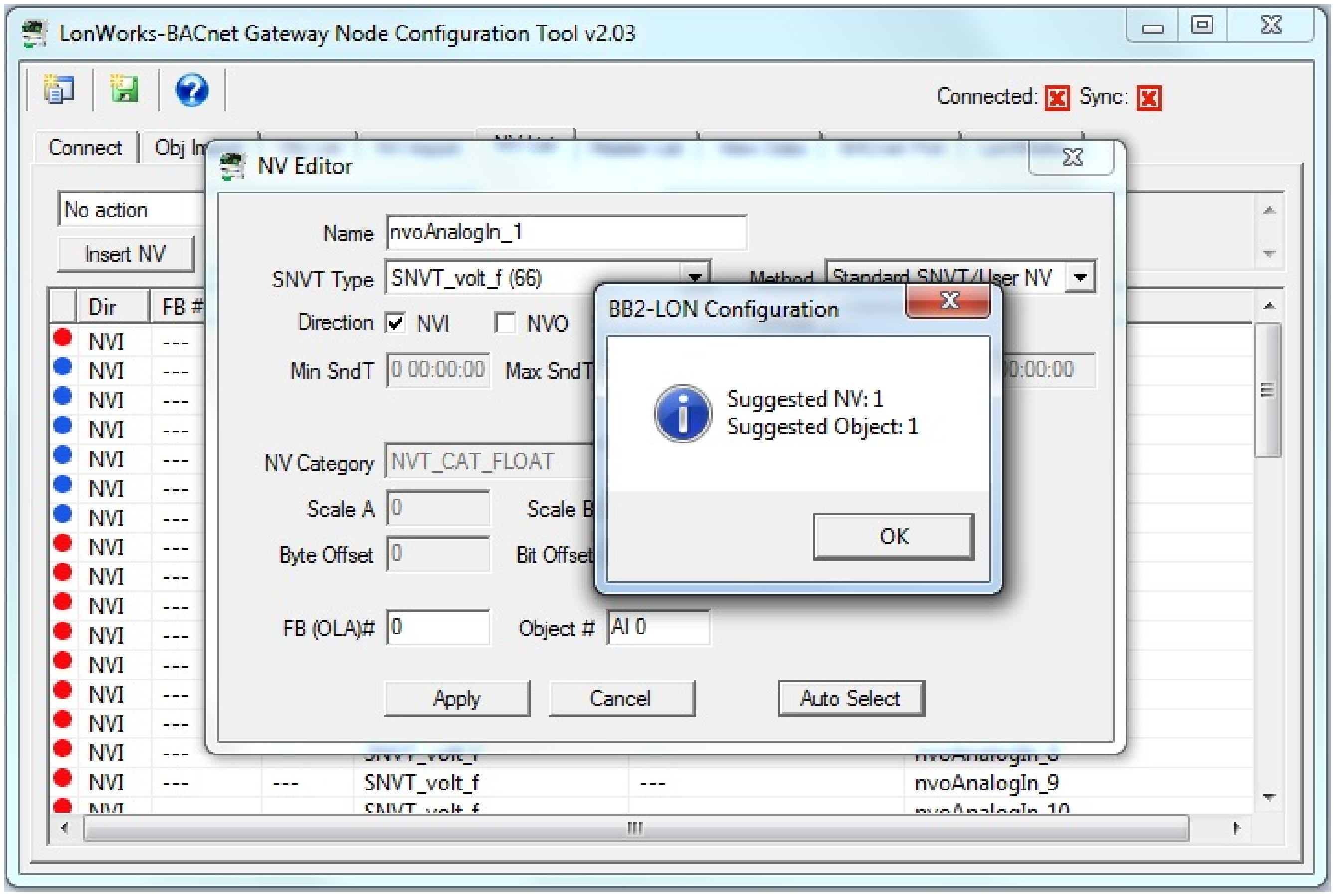
Dir	FB #				
NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_9

To remedy this problem, go back to the Obj List page and click Append Obj a time or two to allocate additional unused objects. Change object types on the Obj List page if applicable.



	FB #	Loc ...	R/W	Device	Mac	Type	Inst	Prop	Index	Bit	Data	Priority	Name
●	---	AI 1	-	0	0	AI	0	0	-1	0	NULL	0	AI 1
●	---	AI 2	-	0	0	AI	0	0	-1	0	NULL	0	AI 2
●	---	AI 3	-	0	0	AI	0	0	-1	0	NULL	0	AI 3
●	---	AI 4	-	0	0	AI	0	0	-1	0	NULL	0	AI 4
●	---	AI 5	-	0	0	AI	0	0	-1	0	NULL	0	AI 5
●	---	AI 6	-	0	0	AI	0	0	-1	0	NULL	0	AI 6

Now, upon returning to the NV Editor, when you click Auto Select, you will get some suggestions. Enter these numbers, and they will become effective upon clicking Apply. Once applied, the FB# and Object# cannot be edited. You can only delete and re-insert.



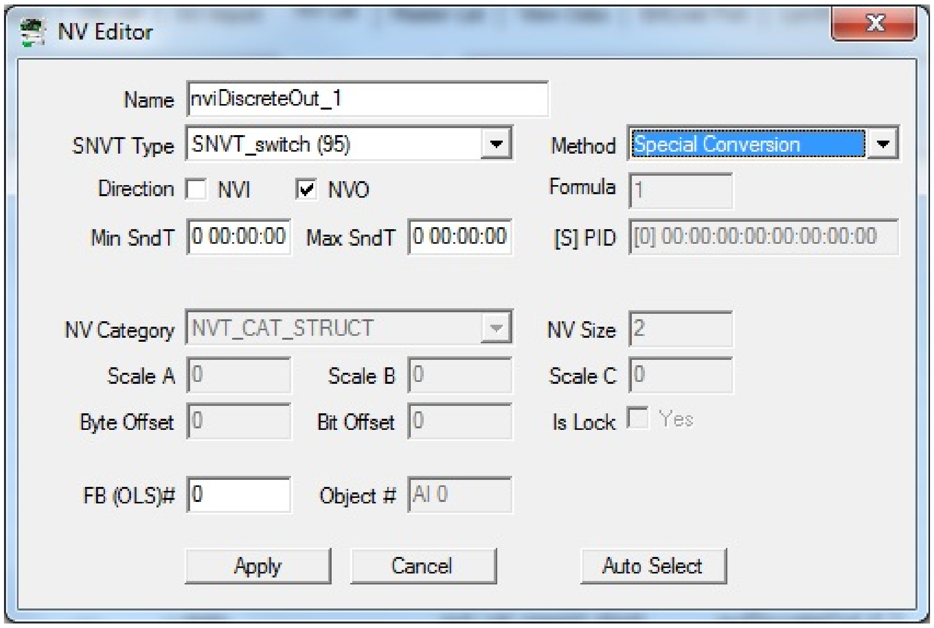
Name: nvoAnalogIn_1
SNVT Type: SNVT_volt_f (66)
Method: Standard SNVT/User NV
Direction: ☑ NVI ☐ NVO
Min SndT: 0 00:00:00
NV Category: NVT_CAT_FLOAT
Scale A: 0
Byte Offset: 0
FB (OLA)#: 0
Object #: AI 0

Suggested NV: 1
Suggested Object: 1

Dir	FB #				
● NVI	---				
● NVI	---				
● NVI	---				
● NVI	---				
● NVI	---				
● NVI	---				
● NVI	---				
● NVI	---				
● NVI	---				
● NVI	---				
● NVI	---				
● NVI	---				
● NVI	---				
● NVI	---				
● NVI	---				
● NVI	---	---	SNVT_volt_f	---	nvoAnalogIn_9

The SNVT_switch example is illustrated below. The LonMark definition of a "switch" actually contains two elements of data, a state and a level (as would be used for a dimmer switch). You cannot correctly control anything via a SNVT_switch without properly dealing with both elements of data. In BACnet terms, this would require two BACnet objects to control one switch, and this is not typically desirable from a BACnet point of view. Therefore, the Babel Buster gateway provides a "special conversion" such that a single BACnet object containing a value from 0 to 100 (implied percent) will result in both parts of the SNVT_switch being set correctly to control on/off (100% or 0%) or any level in between.

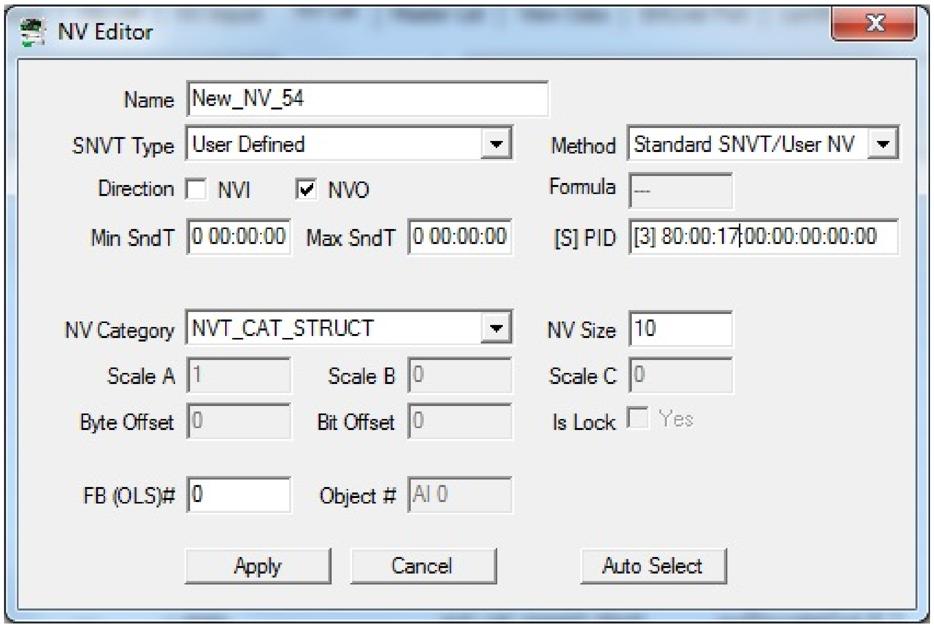
Your selection of whether to use the special conversion will decide whether SNVT_switch maps to one or two BACnet objects, and therefore whether SNVT_switch maps to one or two data objects. For this reason, it is important that you make all of your NV related selections and configurations BEFORE assigning function blocks and data objects.



Name	nviDiscreteOut_1				
SNVT Type	SNVT_switch (95)			Method	Special Conversion
Direction	☐ NVI	☑ NVO		Formula	1
Min SndT	0 00:00:00	Max SndT	0 00:00:00	[S] PID	[0] 00:00:00:00:00:00:00:00
NV Category	NVT_CAT_STRUCT			NV Size	2
Scale A	0	Scale B	0	Scale C	0
Byte Offset	0	Bit Offset	0	Is Lock	☐ Yes
FB (OLS)#	0	Object #	AI 0		

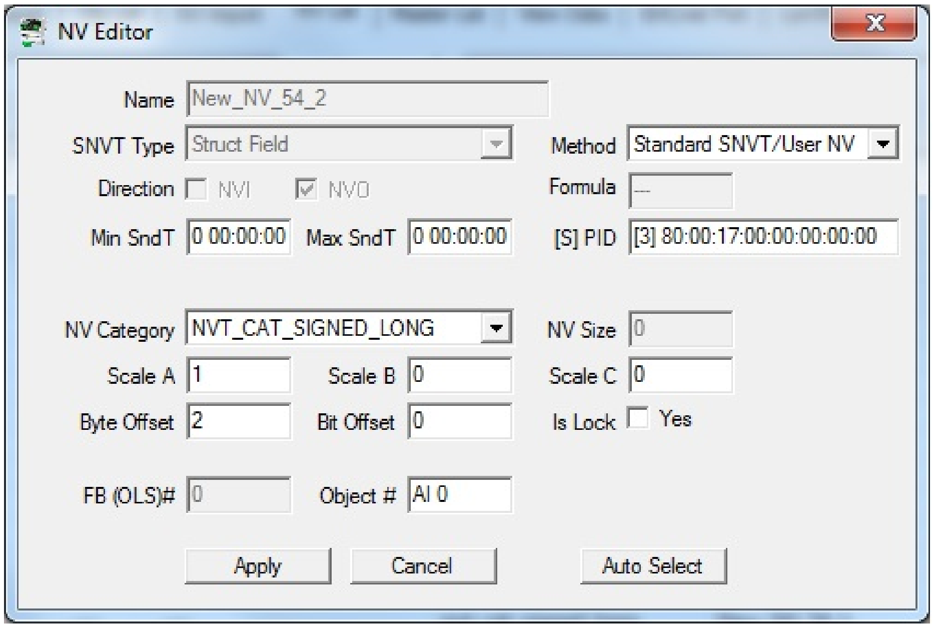
Creating a user defined structured network variable is illustrated below. The parent NV must have SNVT type set to User Defined. The Scope [S] and Program ID PID must be non-zero. Scope of 3 means manufacturer defined. Refer to LonMark standards for additional information about scope values.

The NV Category must be NVT_CAT_STRUCT and the NV Size must be set to the total number of bytes of data that this NV will occupy, including all fields of the structure.



Name	New_NV_54				
SNVT Type	User Defined			Method	Standard SNVT/User NV
Direction	☐ NVI	☑ NVO		Formula	---
Min SndT	0 00:00:00	Max SndT	0 00:00:00	[S] PID	[3] 80:00:17:00:00:00:00:00
NV Category	NVT_CAT_STRUCT			NV Size	10
Scale A	1	Scale B	0	Scale C	0
Byte Offset	0	Bit Offset	0	Is Lock	☐ Yes
FB (OLS)#	0	Object #	AI 0		

For each field of the structured NV, select a standard data bype for NV Category. The first field will have a byte offset of zero. But all subsequent fields will need to have a non-zero offset so that the data is taken from the right place in the structure. When the network variable (NV) is read from the LonWorks device, or sent ot the LonWorks device, it is simply a string of up to 31 bytes of data. Interpreting the bytes correctly requires both a data type, and Byte Offset. Offset of zero is the first byte in the data stream. As an example, if you have two consecutive NVT_CAT_UNSIGNED_LONG fields, the offset for the second one must be 2 since each 'unsigned long' (in LonWorks terms) is 2 bytes. Note that the definition of 'unsigned long' in LonWorks is different than the definition of 'unsigned long' in most computer programming languages such as C.



Name	New_NV_54_2		
SNVT Type	Struct Field	Method	Standard SNVT/User NV
Direction	☐ NVI ☑ NVO	Formula	---
Min SndT	0 00:00:00	Max SndT	0 00:00:00
[S] PID	[3] 80:00:17:00:00:00:00:00		
NV Category	NVT_CAT_SIGNED_LONG	NV Size	0
Scale A	1	Scale B	0
Scale C	0		
Byte Offset	2	Bit Offset	0
Is Lock	☐ Yes		
FB (OLS)#	0	Object #	AI 0

Special conversions available for certain SNVT's are listed below. The converson formula is determined automatically based on the SNVT type to which 'Special Conversion' is applied. The table below lists conversion from NV to internal data object value. The process is reversed for converting internal data object value to a standard NV. The conversion is initally made to floating point, but if the internal data object is configured as integer, then the value will be truncated and retained as an integer. The internal data object value is also the value that will be accessed as a BACnet object.

Special conversion formula 1	SNVT_switch	The structure consisting of state and value is converted to a single floating point value in the range of 0.00% to 100.00%.
Special conversion formula 2	SNVT_elapsed_tm	The structure containing 5 fields from days to millisconds is converted to a single floating point value of seconds.

NV Editor fields are used as follows:

Name	NV name that will be exported to XIF file when saved. Name is limited (LonWorks requirement) to 16 characters and may contain no spaces.
SNVT Type	SNVT type is any of the LonMark standard network variable types. This field may also be set to "User Defined" to work with non-standard variable types including UNVT's or User Network Variable Types.
Direction	Sets Input or Output direction of the Network Variable. This direction setting is only in the NV List, and your selection will be assigned to the appropriate function block type when they are assigned. If an FB# has already been allocated, you cannot change the direction.
Min Snd T	The minimum send time is the minimum amount of time that must elapse between updates of a network variable output (NVO). It does not apply to an NVI. Minimum send time defines the maximum update rate. If set to zero, the feature is disabled. The LonMark standard default is 15 seconds.
Max Snd T	The maximum send time is the maximum amount of time that can elapse between updates. If there has been no data value change to report, the NVO is transmitted anyway when the maximum send time expires. This does not apply to an NVI. If set to zero, the feature is disabled.

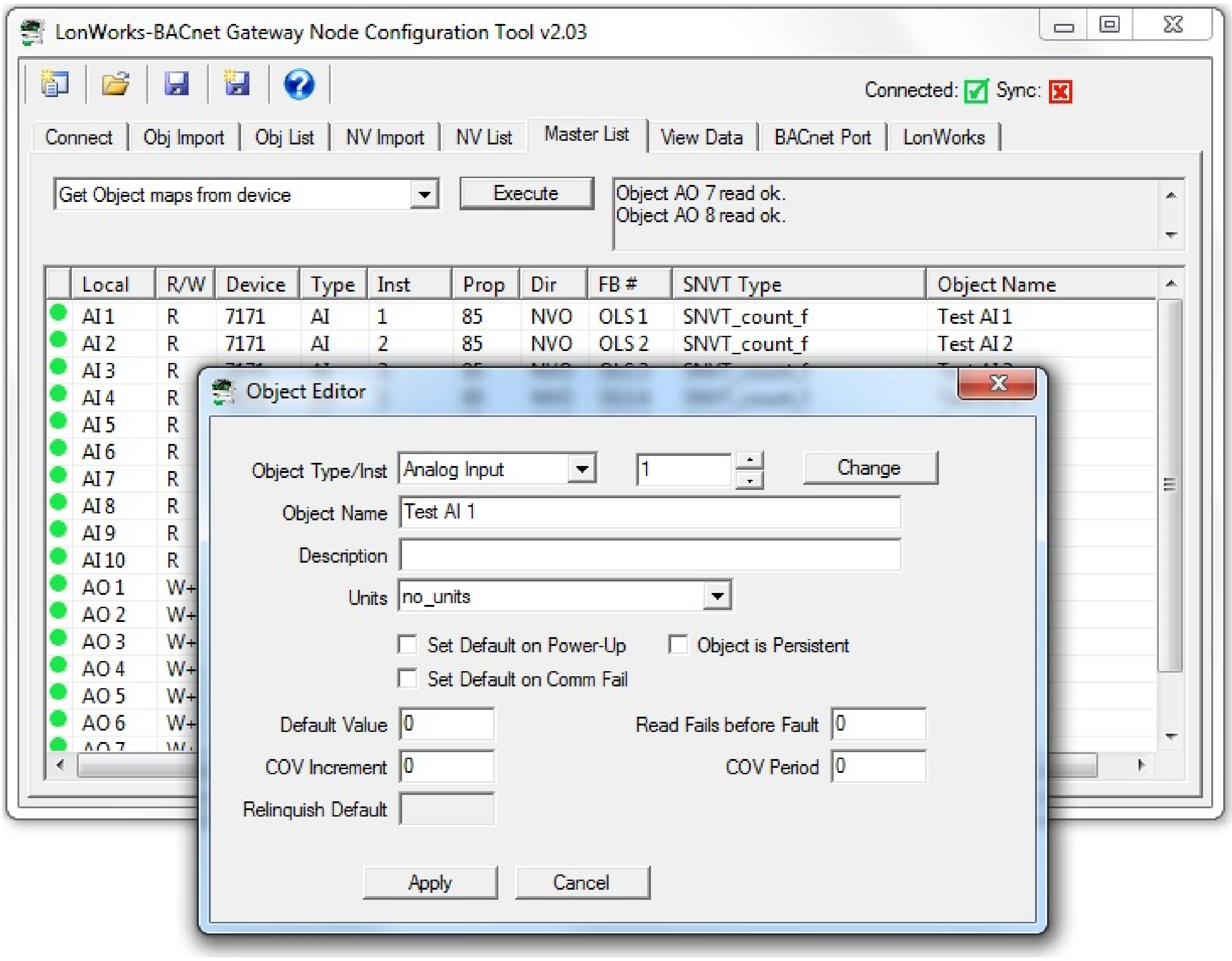
Method	Method will most often be "Standard SNVT/User NV". Other options include "Copy Raw Data" and "Special Conversion". The available special conversions are noted above. "Copy raw" means exactly that - copy bytes of data as they are without any conversion or other treatment.
Formula	Identifies which conversion formula applies when "Special Conversion" is selected as Method.
[S] PID	The [S] is network variable scope. A scope of 0 indicates standard LonMark NV types. A scope of 3 through 6 indicates manufacturer defined NV types. 0 – Standard – applies to all devices. 1 – Reserved for future use. 2 – Reserved for future use. 3 – Manufacturer – applies to all devices from the manufacturer specified in the program ID template. 4 – Manufacturer and Device Class – applies to all devices from the manufacturer with the device class specified in the program ID template. 5 – Manufacturer, Device Class, and Device Subclass – applies to all devices from the manufacturer with the device class and device subclass specified in the program ID template. 6 – Manufacturer, Device Class, Device Subclass, and Device Model – applies to all devices of the specified type and manufacturer specified in the program ID template. The PID, consisting of 8 hexadecimal values, is the Program ID of the device or device class for which the UNVT (User Network Variable Type) applies.
NV Category	NV Category defines the data type for non-structured network variables, or declares the NV to be a structure, or enumeration. When used in a field of a structure, "bitfield" is also applicable. Although "array" and "union" are defined by LonMark, they are not generally used in the definitions of standard NV types, and are not processed by the gateway.
NV Size	NV size is the number of bytes of data provided by this NV. If the NV is structured, this is the number of bytes in the entire structure when looking at the parent NV. This is the number of bytes in just the field if looking at a field in a structure.
Scale A, B, C	The scale values are used to convert raw binary data as transmitted over the wire to engineering units as would be displayed in a user interface. The scale values are fixed and defined by LonMark for Standard Network Variable Types. You need to enter them for user defined types. The scaling formula is S = a*10b*(R+c) where R is raw data, and S is scaled data.
Byte Offset	Byte offset is only used in structured network variables, and defines, for each field of the structure, where in the string of data bytes this field begins.
Bit Offset	Bit offset is used to select a specific bit when the NV Category is "bitfield". Byte offset is used to specify which byte within a structure the bit should be taken from. Bit offset is 0 to 7, with bit offset of 0 being the most significant bit in the byte. Note that this is backwards from most interpretations of "bit 0". LonWorks interpretation of bit is bit offset from the start of the byte, and start of byte is interpreted as most significant bit.
Is Lock	The Lock applies only to structured variables. When a structured NVO maps to multiple BACnet objects, the NVO will be updated (transmitted on the LonWorks network) any time any part of the structure is updated from the BACnet side. If it is desired that the NVO should only be transmitted as a result of update to a specific field (allowing all parts of the structure to be updated before transmitted), then you want to specify one field of the structure as the "lock". The NVO will only

	be transmitted on the LonWorks network when this field is updated from the BACnet side.
FB #	The function block number that has been assigneed will be indicated here, or will be zero if not yet assigned. The direction (NVO or NVI) will determine whether the FB is indicated as OLS (Open Loop Sensor) or OLA (Open Loop Actuator).
Object #	The internal data object number that has been assigned will be indicated here, or will be zero if not yet assigned. This object number is where you will find mapping for the BACnet object associated with the displayed NV entry.

9 Tool 'Master List' Page

9.1 Editing Configuration from Master List Page

Clicking on the Object number column will open the Object Editor dialog as illustrated below. This is the same editor described in Section 6 of this user guide. Refer to that section for details about the BACnet Object Editor.


LonWorks-BACnet Gateway Node Configuration Tool v2.03

Connected: ✓ Sync: ✗

Connect | Obj Import | Obj List | NV Import | NV List | Master List | View Data | BACnet Port | LonWorks

Get Object maps from device | Execute

```
Object AO 7 read ok.
Object AO 8 read ok.
```

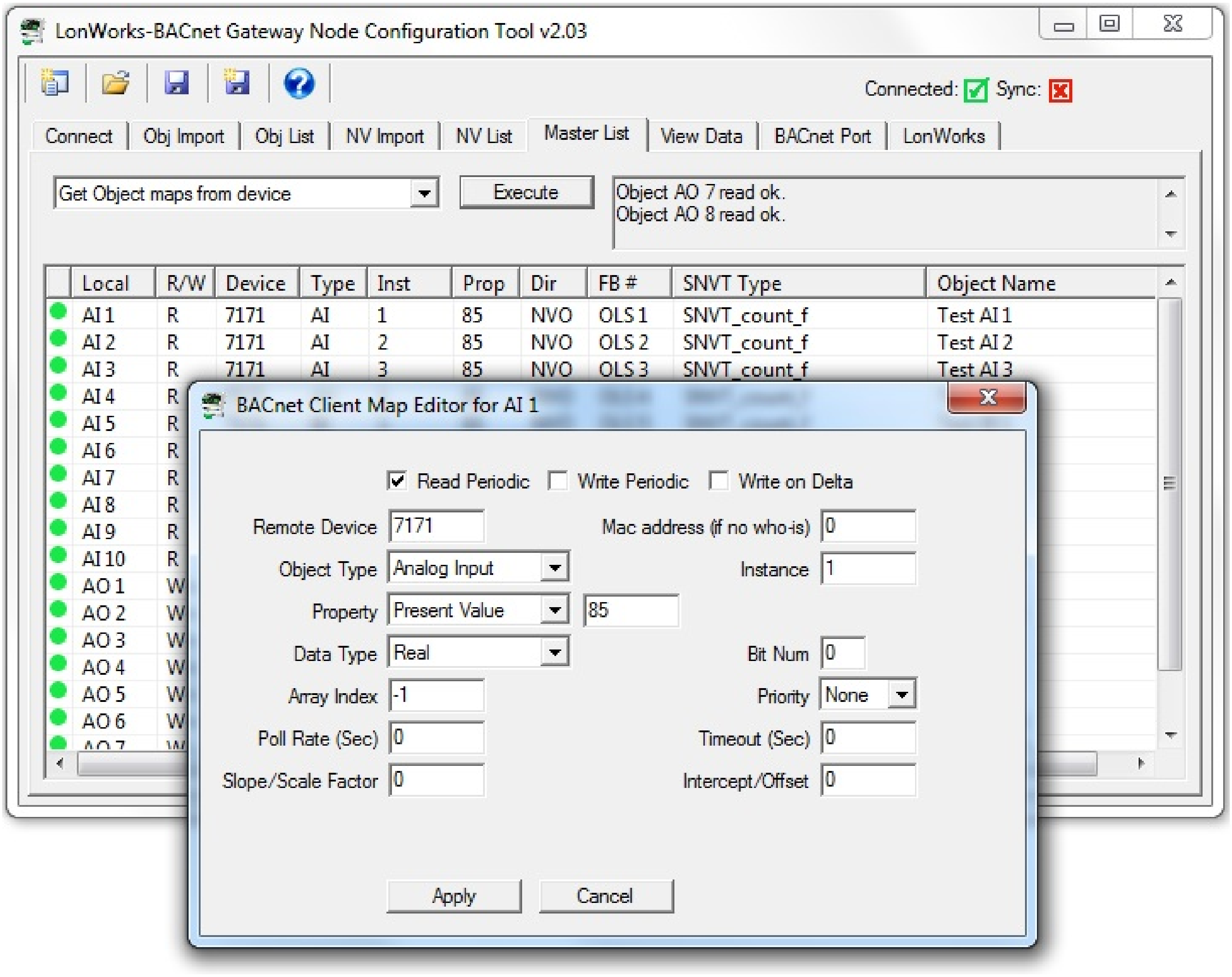
	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	SNVT Type	Object Name
●	AI 1	R	7171	AI	1	85	NVO	OLS 1	SNVT_count_f	Test AI 1
●	AI 2	R	7171	AI	2	85	NVO	OLS 2	SNVT_count_f	Test AI 2
●	AI 3	R								
●	AI 4	R								
●	AI 5	R								
●	AI 6	R								
●	AI 7	R								
●	AI 8	R								
●	AI 9	R								
●	AI 10	R								
●	AO 1	W+								
●	AO 2	W+								
●	AO 3	W+								
●	AO 4	W+								
●	AO 5	W+								
●	AO 6	W+								

Object Editor

Object Type/Inst	Analog Input, 1 (Change)
Object Name	Test AI 1
Description	
Units	no_units
Set Default on Power-Up	☐
Object is Persistent	☐
Set Default on Comm Fail	☐
Default Value	0
Read Fails before Fault	0
COV Increment	0
COV Period	0
Relinquish Default	

Apply | Cancel

Clicking on the client map area of a line on the Master List will open the BACnet Client Map Editor dialog as illustrated below. This is the same editor described in Section 6 of this user guide. Refer to that section for details about the BACnet Client Map Editor.



Get Object maps from device

Object AO 7 read ok.
Object AO 8 read ok.

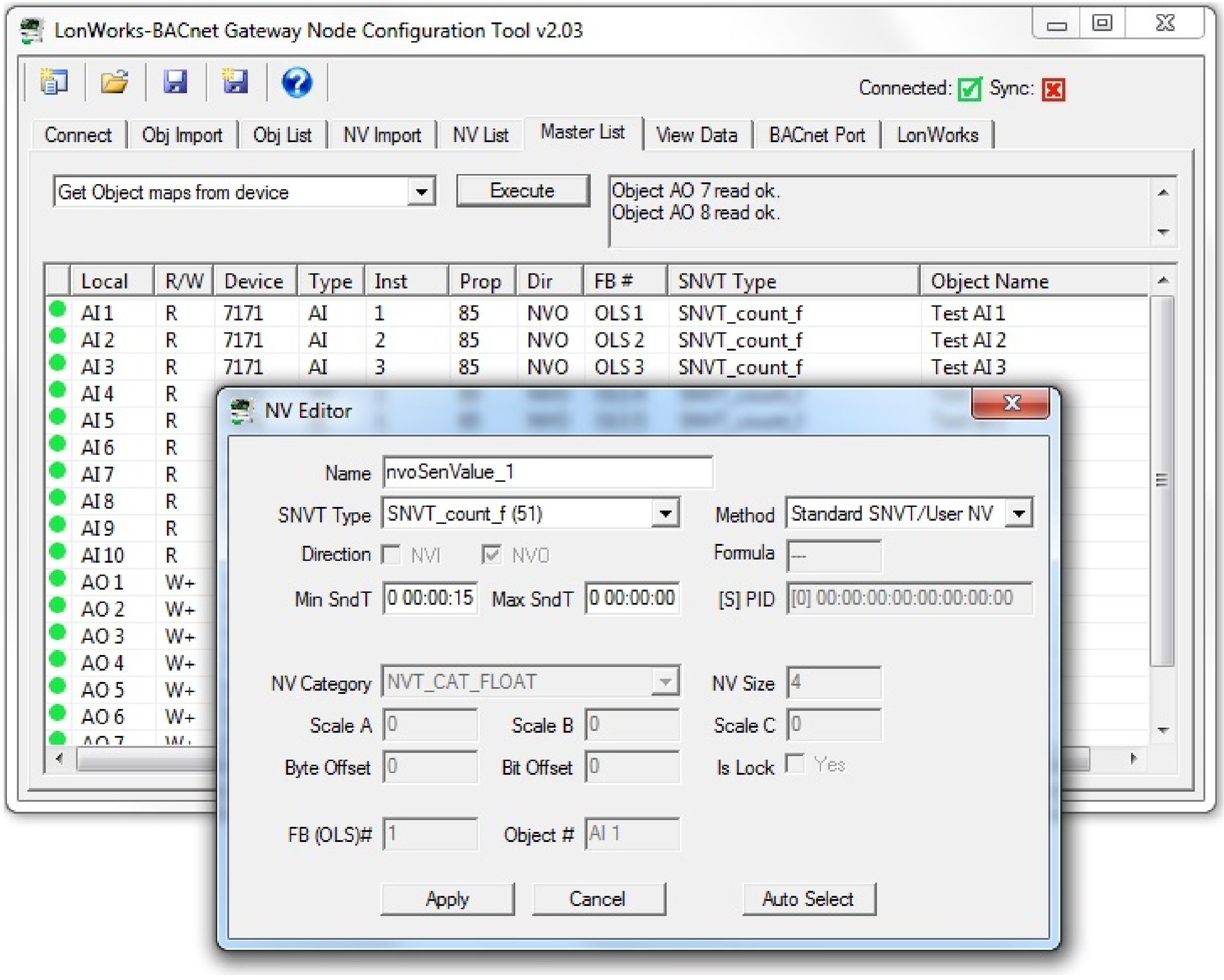
	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	SNVT Type	Object Name
●	AI 1	R	7171	AI	1	85	NVO	OLS 1	SNVT_count_f	Test AI 1
●	AI 2	R	7171	AI	2	85	NVO	OLS 2	SNVT_count_f	Test AI 2
●	AI 3	R	7171	AI	3	85	NVO	OLS 3	SNVT_count_f	Test AI 3
●	AI 4	R								
●	AI 5	R								
●	AI 6	R								
●	AI 7	R								
●	AI 8	R								
●	AI 9	R								
●	AI 10	R								
●	AO 1	W								
●	AO 2	W								
●	AO 3	W								
●	AO 4	W								
●	AO 5	W								
●	AO 6	W								

BACnet Client Map Editor for AI 1

[✓] Read Periodic ☐ Write Periodic ☐ Write on Delta

Remote Device	7171	Mac address (if no who-is)	0
Object Type	Analog Input	Instance	1
Property	Present Value 85		
Data Type	Real	Bit Num	0
Array Index	-1	Priority	None
Poll Rate (Sec)	0	Timeout (Sec)	0
Slope/Scale Factor	0	Intercept/Offset	0

Clicking on the Network Variable area of a line on the Master List will open the NV Editor dialog as illustrated below. This is the same editor described in Section 8 of this user guide. Refer to that section for details about the NV Editor.



Get Object maps from device

```
Object AO 7 read ok.
Object AO 8 read ok.
```

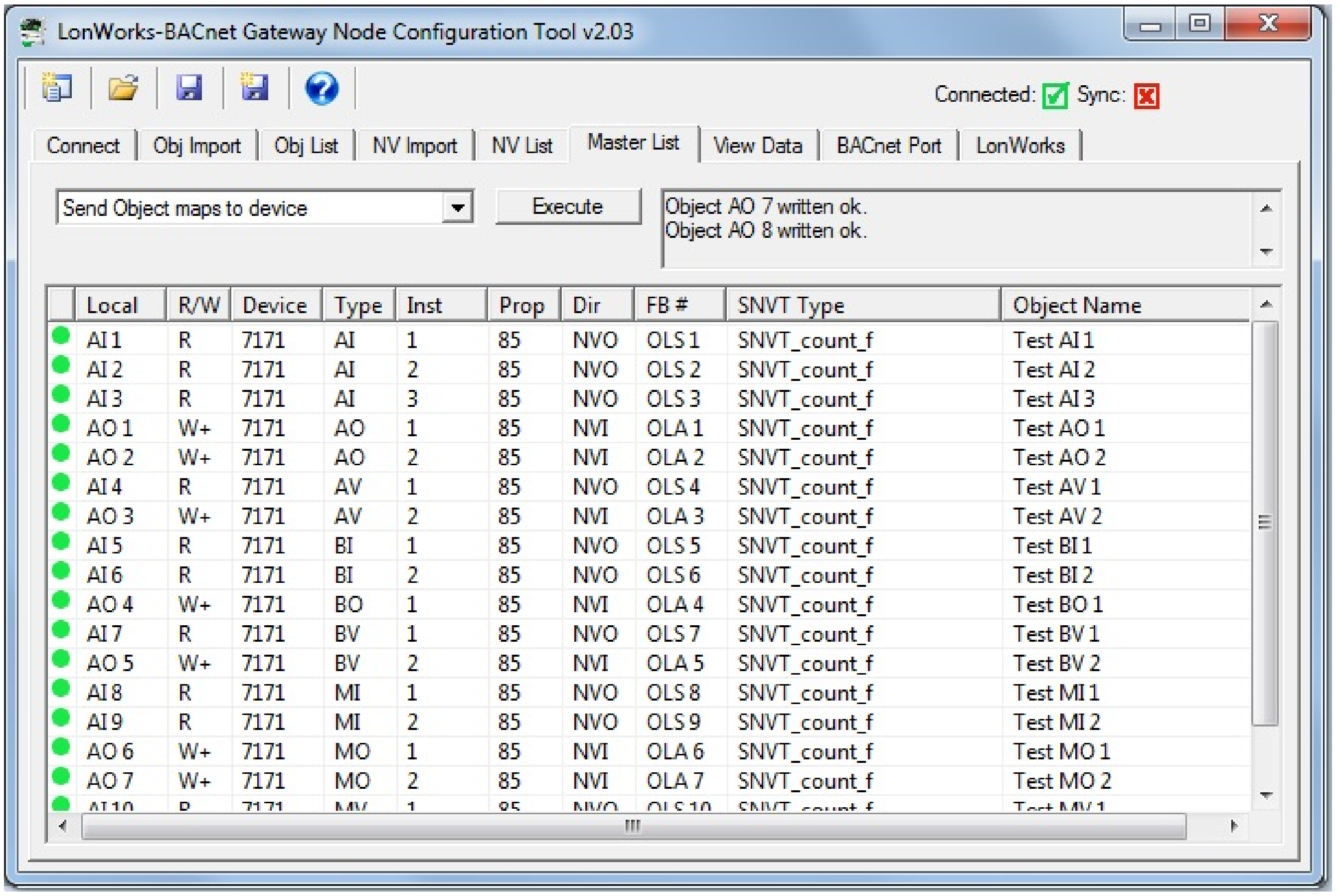
	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	SNVT Type	Object Name
●	AI 1	R	7171	AI	1	85	NVO	OLS 1	SNVT_count_f	Test AI 1
●	AI 2	R	7171	AI	2	85	NVO	OLS 2	SNVT_count_f	Test AI 2
●	AI 3	R	7171	AI	3	85	NVO	OLS 3	SNVT_count_f	Test AI 3
●	AI 4	R								
●	AI 5	R								
●	AI 6	R								
●	AI 7	R								
●	AI 8	R								
●	AI 9	R								
●	AI 10	R								
●	AO 1	W+								
●	AO 2	W+								
●	AO 3	W+								
●	AO 4	W+								
●	AO 5	W+								
●	AO 6	W+								

Name	nvoSenValue_1				
SNVT Type	SNVT_count_f (51)			Method	Standard SNVT/User NV
Direction	NVI	[✓ NVO]		Formula	---
Min SndT	0 00:00:15	Max SndT	0 00:00:00	[S] PID	[0] 00:00:00:00:00:00:00:00
NV Category	NVT_CAT_FLOAT			NV Size	4
Scale A	0	Scale B	0	Scale C	0
Byte Offset	0	Bit Offset	0	Is Lock	Yes
FB (OLS)#	1	Object #	AI 1		

9.2 Sending Configuration To Device

Select "Send Object maps to device" to send Object configuration to the gateway. You must first "Send NV definitions to device". There are a total of four pages containing information that must be sent to the gateway to fully configure it.

- NV List (or Master List): Send NV definitions to device
- Master List: Send Object maps to device
- BACnet Port: Use 'Change' to set device info, use 'Reconfigure' if object counts need changing (Get first to see)
- LonWorks: Set Location



Object AO 7 written ok.
Object AO 8 written ok.

	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	SNVT Type	Object Name
●	AI 1	R	7171	AI	1	85	NVO	OLS 1	SNVT_count_f	Test AI 1
●	AI 2	R	7171	AI	2	85	NVO	OLS 2	SNVT_count_f	Test AI 2
●	AI 3	R	7171	AI	3	85	NVO	OLS 3	SNVT_count_f	Test AI 3
●	AO 1	W+	7171	AO	1	85	NVI	OLA 1	SNVT_count_f	Test AO 1
●	AO 2	W+	7171	AO	2	85	NVI	OLA 2	SNVT_count_f	Test AO 2
●	AI 4	R	7171	AV	1	85	NVO	OLS 4	SNVT_count_f	Test AV 1
●	AO 3	W+	7171	AV	2	85	NVI	OLA 3	SNVT_count_f	Test AV 2
●	AI 5	R	7171	BI	1	85	NVO	OLS 5	SNVT_count_f	Test BI 1
●	AI 6	R	7171	BI	2	85	NVO	OLS 6	SNVT_count_f	Test BI 2
●	AO 4	W+	7171	BO	1	85	NVI	OLA 4	SNVT_count_f	Test BO 1
●	AI 7	R	7171	BV	1	85	NVO	OLS 7	SNVT_count_f	Test BV 1
●	AO 5	W+	7171	BV	2	85	NVI	OLA 5	SNVT_count_f	Test BV 2
●	AI 8	R	7171	MI	1	85	NVO	OLS 8	SNVT_count_f	Test MI 1
●	AI 9	R	7171	MI	2	85	NVO	OLS 9	SNVT_count_f	Test MI 2
●	AO 6	W+	7171	MO	1	85	NVI	OLA 6	SNVT_count_f	Test MO 1
●	AO 7	W+	7171	MO	2	85	NVI	OLA 7	SNVT_count_f	Test MO 2
●	AI 10	R	7171	MV	1	85	NVO	OLS 10	SNVT_count_f	Test MV 1

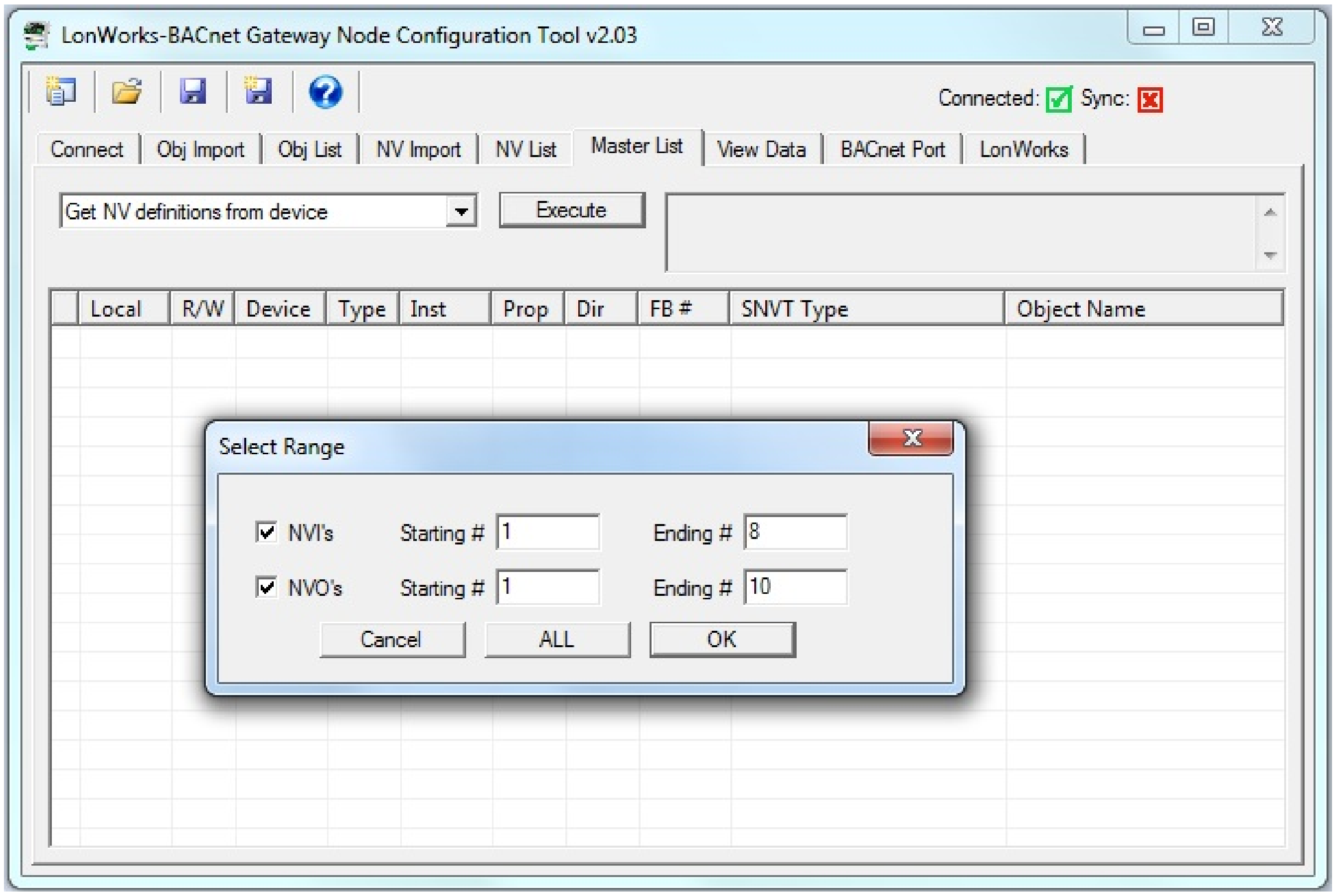
9.3 Getting Configuration From Device

The configuration tool can be used to retrieve all necessary information from a previously configured device. Once retrieved, you can use that information to replicate the same configuration in another device, or simply save the configuration to a file for later use.

There are a total of four pages containing information that must be read from the gateway to obtain the complete configuration.

- NV List (or Master List): Get NV definitions from device
- Master List: Get Object maps from device (should Get Counts first)
- BACnet Port: Get Info and Get Counts
- LonWorks: Get Location

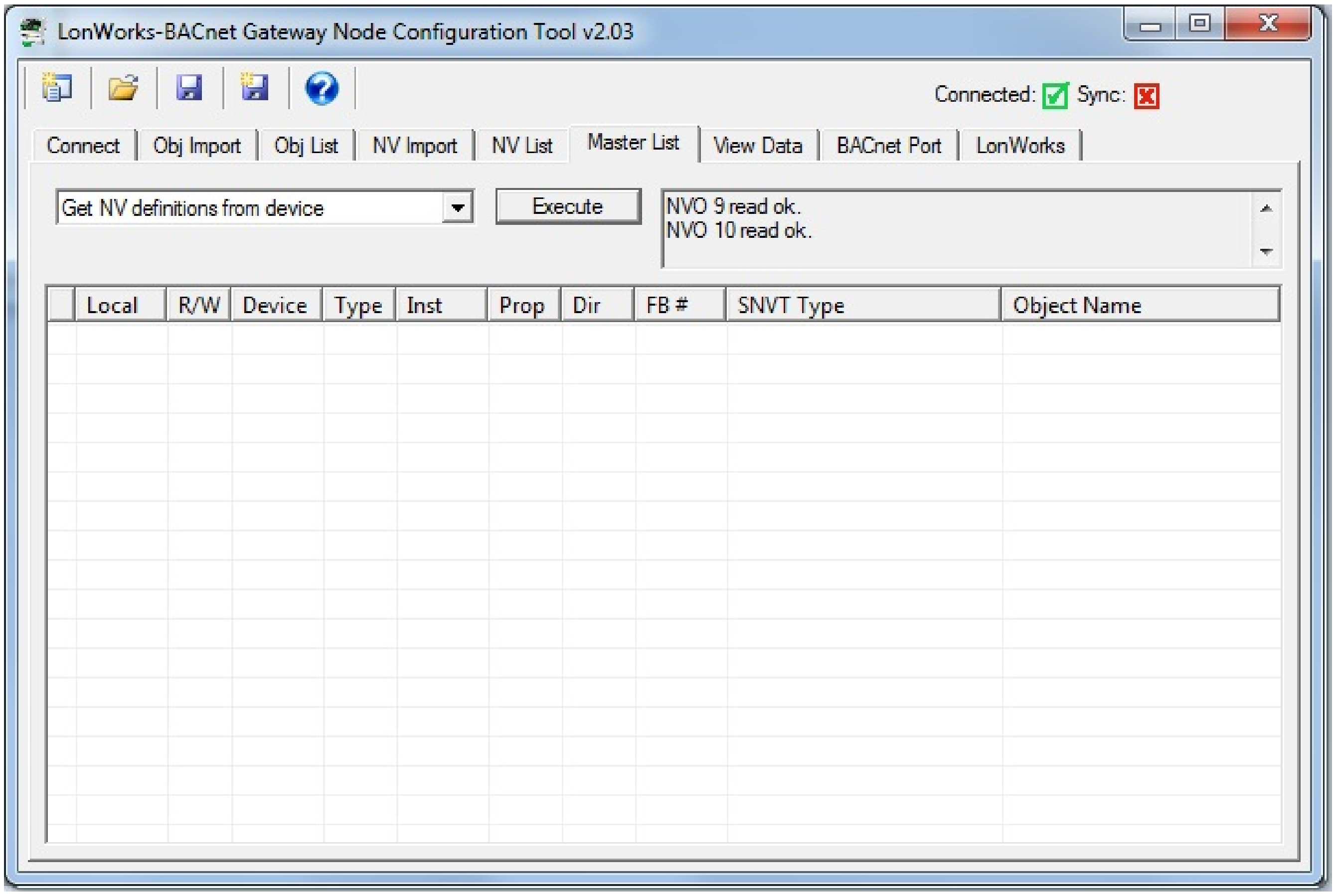
Upon clicking Execute with "Get NV definitions from device" selected, the tool will quiery the gateway to see how many NVs are actually in use. The range will appear in the dialog, allowing you to get only those NVs that are in use rather than all 240 of them.



Get NV definitions from device

✓	NVI's	Starting # 1	Ending # 8
✓	NVO's	Starting # 1	Ending # 10

Upon completion of reading the list, the NVs will not show up in the Master List, but will show up in the NV List. You need to also read the objects before they will show up in the Master List.

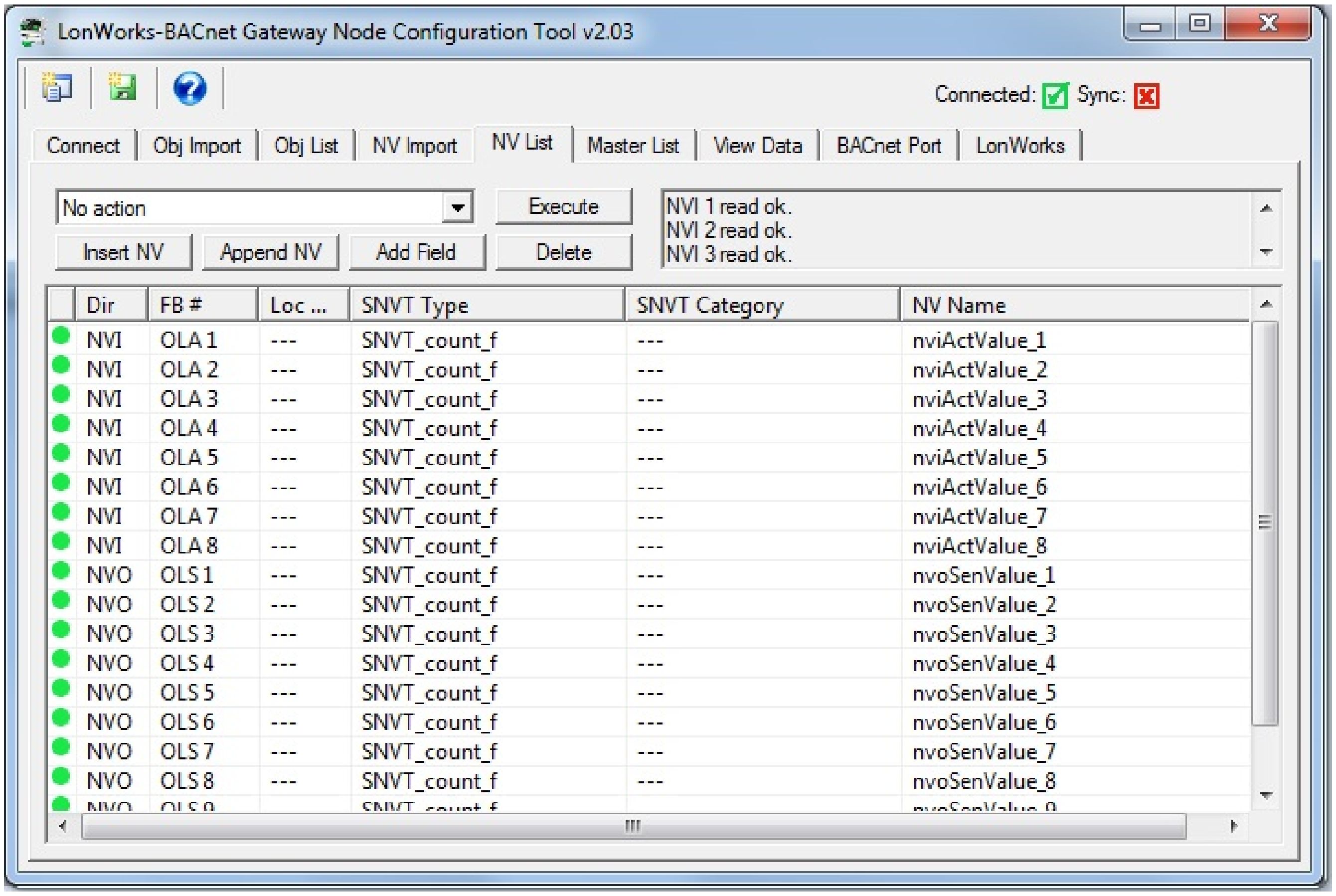


Get NV definitions from device

```
NVO 9 read ok.
NVO 10 read ok.
```

	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	SNVT Type	Object Name

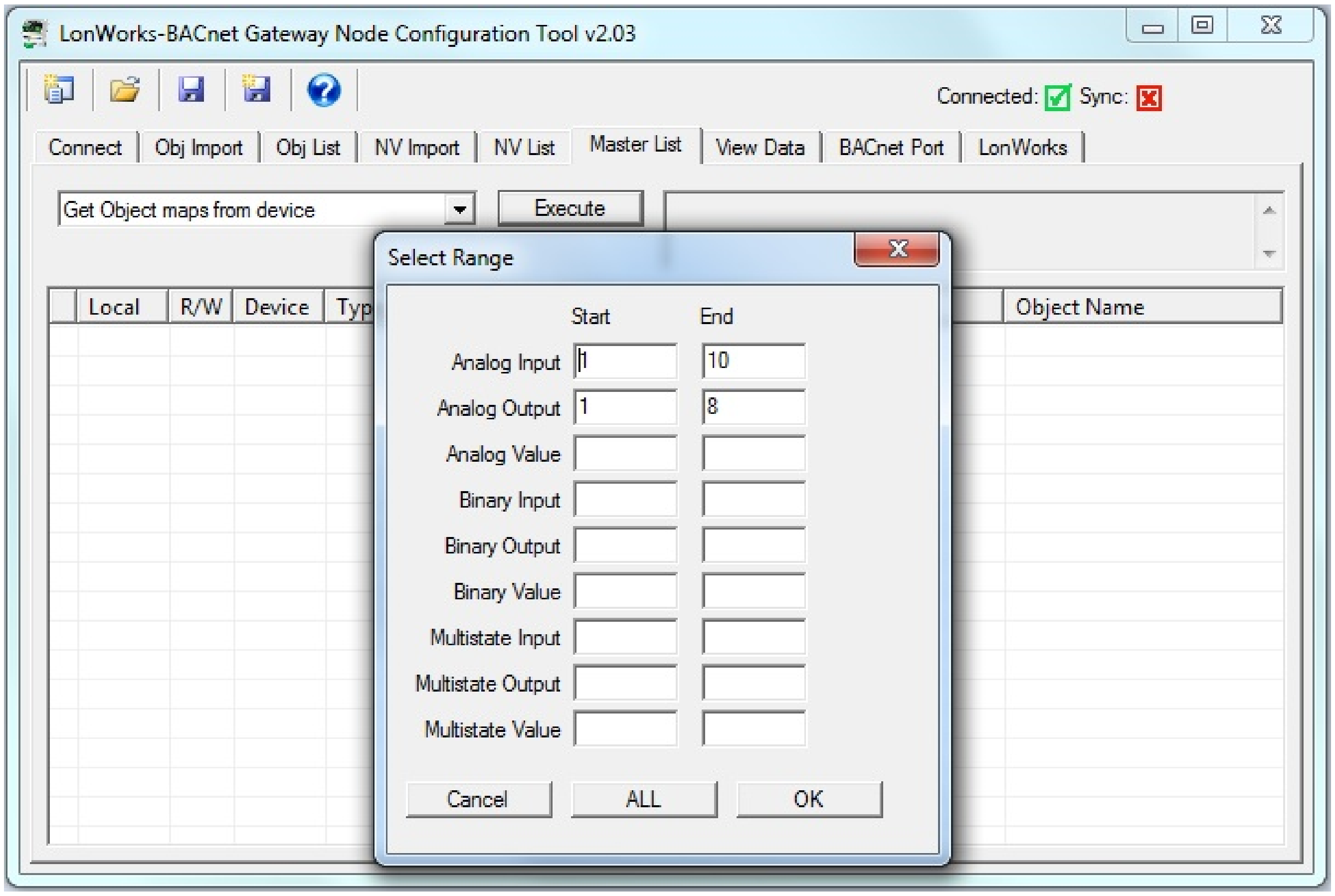
The NV List will appear something like the screen below following the Get NVs. Note that no local objects are yet known.



```
NVI 1 read ok.
NVI 2 read ok.
NVI 3 read ok.
```

	Dir	FB #	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVI	OLA 1	---	SNVT_count_f	---	nviActValue_1
●	NVI	OLA 2	---	SNVT_count_f	---	nviActValue_2
●	NVI	OLA 3	---	SNVT_count_f	---	nviActValue_3
●	NVI	OLA 4	---	SNVT_count_f	---	nviActValue_4
●	NVI	OLA 5	---	SNVT_count_f	---	nviActValue_5
●	NVI	OLA 6	---	SNVT_count_f	---	nviActValue_6
●	NVI	OLA 7	---	SNVT_count_f	---	nviActValue_7
●	NVI	OLA 8	---	SNVT_count_f	---	nviActValue_8
●	NVO	OLS 1	---	SNVT_count_f	---	nvoSenValue_1
●	NVO	OLS 2	---	SNVT_count_f	---	nvoSenValue_2
●	NVO	OLS 3	---	SNVT_count_f	---	nvoSenValue_3
●	NVO	OLS 4	---	SNVT_count_f	---	nvoSenValue_4
●	NVO	OLS 5	---	SNVT_count_f	---	nvoSenValue_5
●	NVO	OLS 6	---	SNVT_count_f	---	nvoSenValue_6
●	NVO	OLS 7	---	SNVT_count_f	---	nvoSenValue_7
●	NVO	OLS 8	---	SNVT_count_f	---	nvoSenValue_8
●	NVO	OLS 9		SNVT_count_f		nvoSenValue_9

Upon clicking Execute with "Get Object maps from device" selected, the tool will quiery the gateway to see how many objects of each type are actually in use. The range will appear in the dialog, allowing you to get only those objects that are in use rather than all 300 of them.


LonWorks-BACnet Gateway Node Configuration Tool v2.03

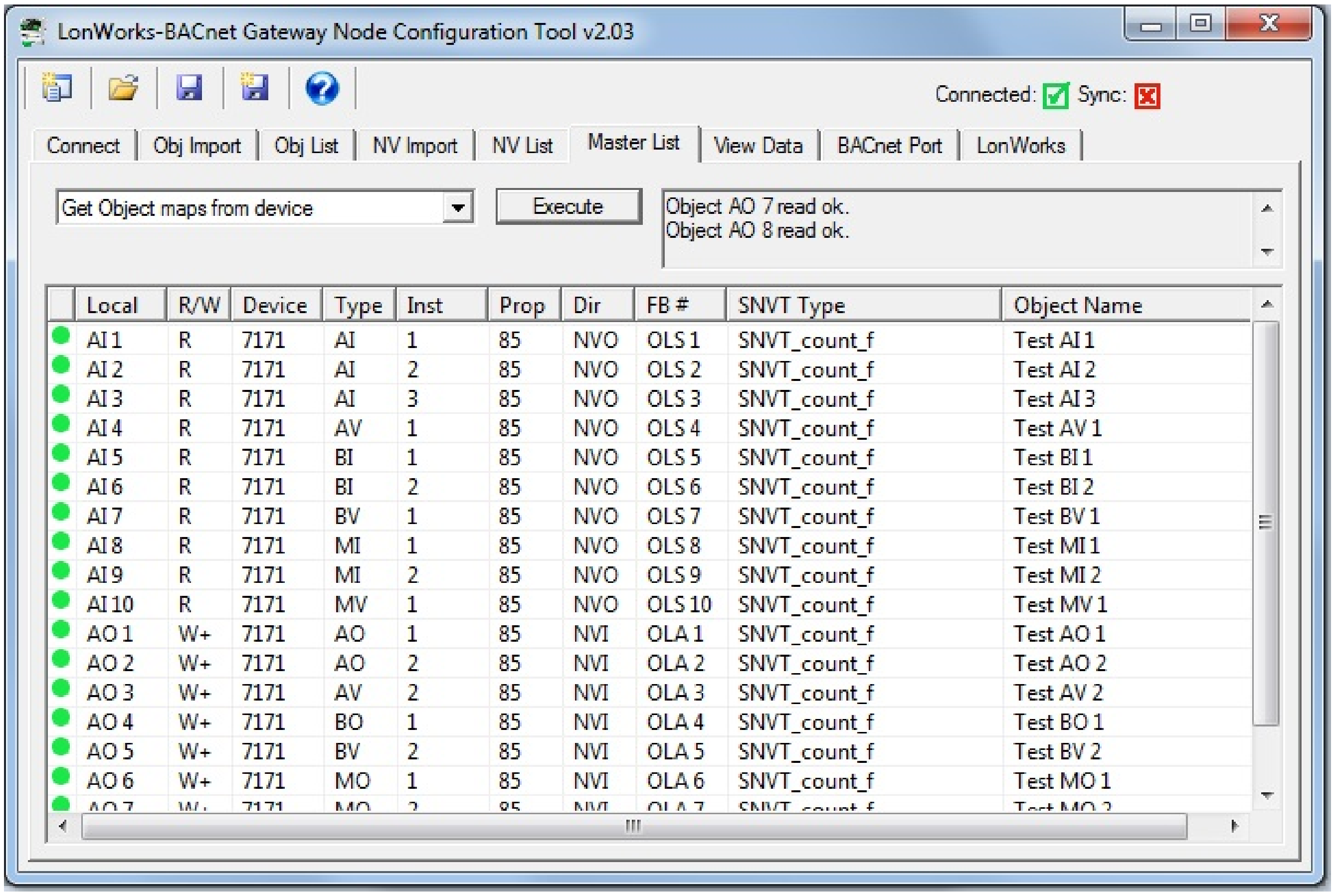
Connected: ✓ Sync: ✗

Get Object maps from device

Select Range

	Start	End
Analog Input	1	10
Analog Output	1	8
Analog Value		
Binary Input		
Binary Output		
Binary Value		
Multistate Input		
Multistate Output		
Multistate Value		

Upon completion of reading the list, the Master List will now be fully populated.



```
Object AO 7 read ok.
Object AO 8 read ok.
```

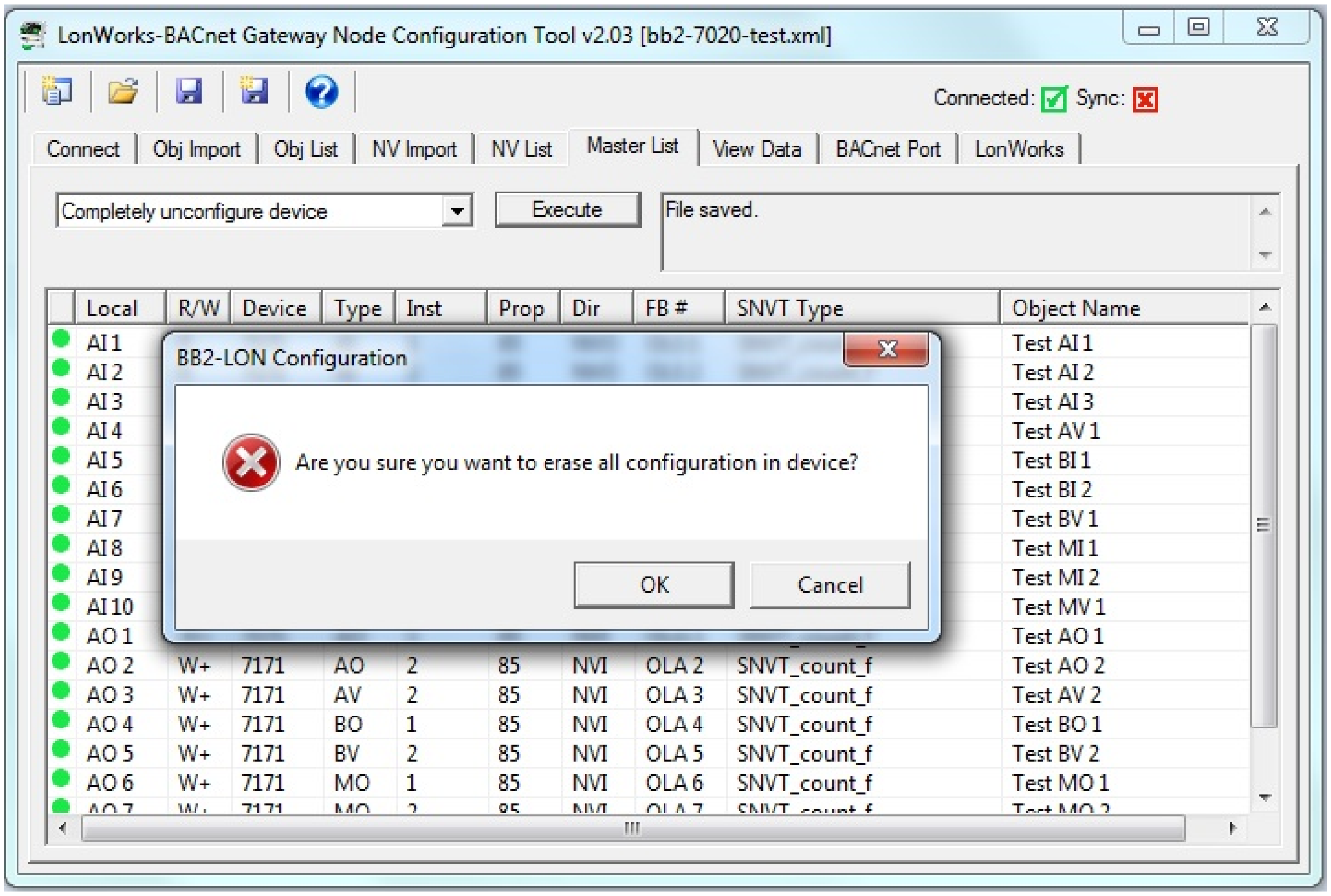
	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	SNVT Type	Object Name
●	AI 1	R	7171	AI	1	85	NVO	OLS 1	SNVT_count_f	Test AI 1
●	AI 2	R	7171	AI	2	85	NVO	OLS 2	SNVT_count_f	Test AI 2
●	AI 3	R	7171	AI	3	85	NVO	OLS 3	SNVT_count_f	Test AI 3
●	AI 4	R	7171	AV	1	85	NVO	OLS 4	SNVT_count_f	Test AV 1
●	AI 5	R	7171	BI	1	85	NVO	OLS 5	SNVT_count_f	Test BI 1
●	AI 6	R	7171	BI	2	85	NVO	OLS 6	SNVT_count_f	Test BI 2
●	AI 7	R	7171	BV	1	85	NVO	OLS 7	SNVT_count_f	Test BV 1
●	AI 8	R	7171	MI	1	85	NVO	OLS 8	SNVT_count_f	Test MI 1
●	AI 9	R	7171	MI	2	85	NVO	OLS 9	SNVT_count_f	Test MI 2
●	AI 10	R	7171	MV	1	85	NVO	OLS 10	SNVT_count_f	Test MV 1
●	AO 1	W+	7171	AO	1	85	NVI	OLA 1	SNVT_count_f	Test AO 1
●	AO 2	W+	7171	AO	2	85	NVI	OLA 2	SNVT_count_f	Test AO 2
●	AO 3	W+	7171	AV	2	85	NVI	OLA 3	SNVT_count_f	Test AV 2
●	AO 4	W+	7171	BO	1	85	NVI	OLA 4	SNVT_count_f	Test BO 1
●	AO 5	W+	7171	BV	2	85	NVI	OLA 5	SNVT_count_f	Test BV 2
●	AO 6	W+	7171	MO	1	85	NVI	OLA 6	SNVT_count_f	Test MO 1
●	AO 7	W+	7171	MO	2	85	NVI	OLA 7	SNVT_count_f	Test MO 2

9.4 Fixing Conflicts

You may run into the error message "Conflicting parameters" if making configuration changes to a gateway previously configured. The "Conflicting parameters" means that the attempted object configuration references a network variable that is not configured the way the object is set to expect.

The easiest way to resolve configuration conflicts is to completely clear the device configuration and resend network variables, followed by object maps. When you execute "Completely unconfigure device", there will be a delay of typically 30 seconds while the device reprograms all of its internal non-volatile configuration memory, restoring it to an 'unused' state. Once complete, proceed with sending configuration to the device as noted above.

You will be prompted with a last chance to cancel the erasing of all configuration.



Completely unconfigure device

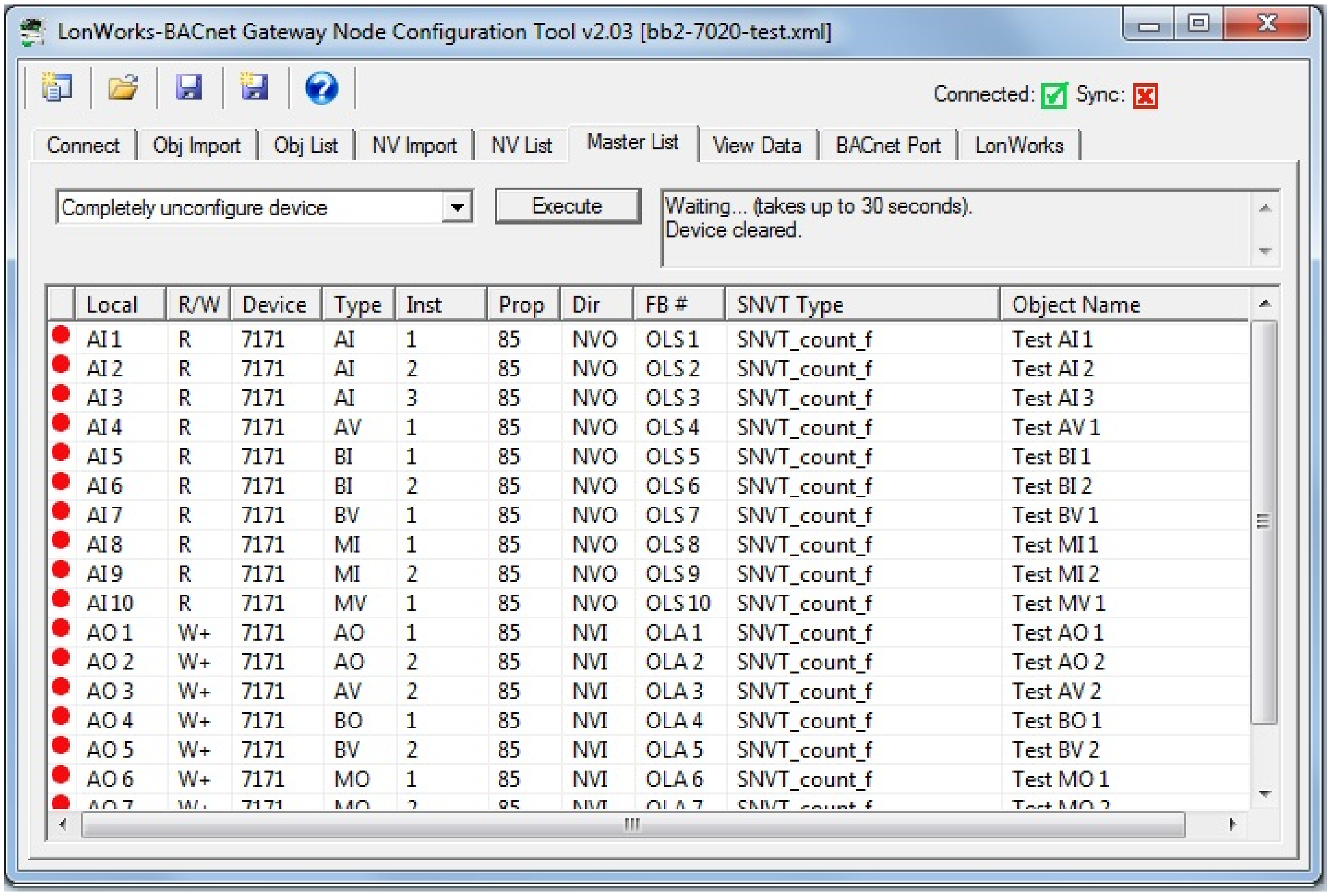
File saved.

BB2-LON Configuration

Are you sure you want to erase all configuration in device?

	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	SNVT Type	Object Name
●	AI 1									Test AI 1
●	AI 2									Test AI 2
●	AI 3									Test AI 3
●	AI 4									Test AV 1
●	AI 5									Test BI 1
●	AI 6									Test BI 2
●	AI 7									Test BV 1
●	AI 8									Test MI 1
●	AI 9									Test MI 2
●	AI 10									Test MV 1
●	AO 1									Test AO 1
●	AO 2	W+	7171	AO	2	85	NVI	OLA 2	SNVT_count_f	Test AO 2
●	AO 3	W+	7171	AV	2	85	NVI	OLA 3	SNVT_count_f	Test AV 2
●	AO 4	W+	7171	BO	1	85	NVI	OLA 4	SNVT_count_f	Test BO 1
●	AO 5	W+	7171	BV	2	85	NVI	OLA 5	SNVT_count_f	Test BV 2
●	AO 6	W+	7171	MO	1	85	NVI	OLA 6	SNVT_count_f	Test MO 1
●	AO 7	W+	7171	MO	2	85	NVI	OLA 7	SNVT_count_f	Test MO 2

You will see the message "Device cleared" when finished. Note that the configuration held by the tool is not cleared by this operation. You can now re-send the configuration from the tool to the device.


LonWorks-BACnet Gateway Node Configuration Tool v2.03 [bb2-7020-test.xml]

Waiting... (takes up to 30 seconds).
Device cleared.

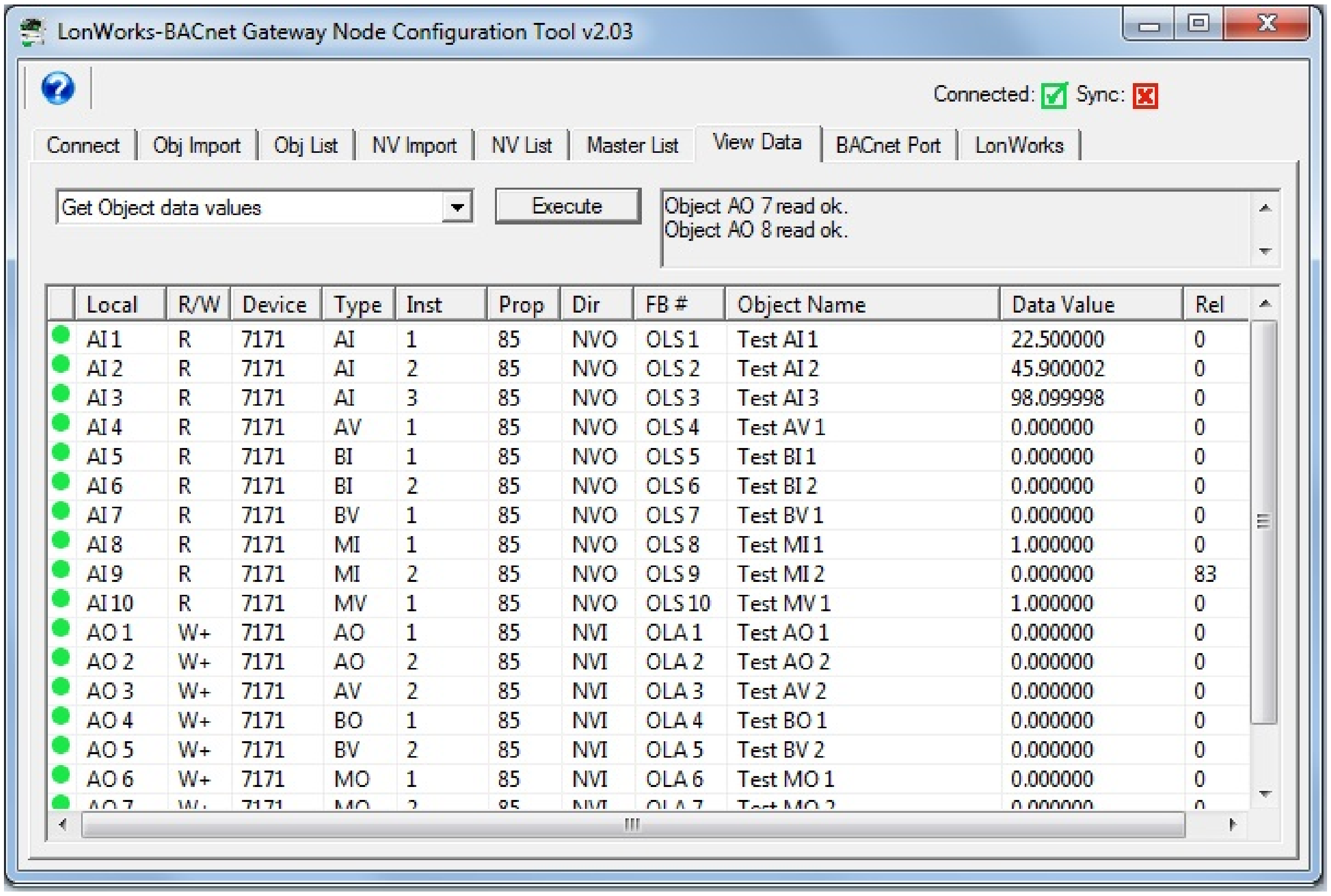
	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	SNVT Type	Object Name
●	AI 1	R	7171	AI	1	85	NVO	OLS 1	SNVT_count_f	Test AI 1
●	AI 2	R	7171	AI	2	85	NVO	OLS 2	SNVT_count_f	Test AI 2
●	AI 3	R	7171	AI	3	85	NVO	OLS 3	SNVT_count_f	Test AI 3
●	AI 4	R	7171	AV	1	85	NVO	OLS 4	SNVT_count_f	Test AV 1
●	AI 5	R	7171	BI	1	85	NVO	OLS 5	SNVT_count_f	Test BI 1
●	AI 6	R	7171	BI	2	85	NVO	OLS 6	SNVT_count_f	Test BI 2
●	AI 7	R	7171	BV	1	85	NVO	OLS 7	SNVT_count_f	Test BV 1
●	AI 8	R	7171	MI	1	85	NVO	OLS 8	SNVT_count_f	Test MI 1
●	AI 9	R	7171	MI	2	85	NVO	OLS 9	SNVT_count_f	Test MI 2
●	AI 10	R	7171	MV	1	85	NVO	OLS 10	SNVT_count_f	Test MV 1
●	AO 1	W+	7171	AO	1	85	NVI	OLA 1	SNVT_count_f	Test AO 1
●	AO 2	W+	7171	AO	2	85	NVI	OLA 2	SNVT_count_f	Test AO 2
●	AO 3	W+	7171	AV	2	85	NVI	OLA 3	SNVT_count_f	Test AV 2
●	AO 4	W+	7171	BO	1	85	NVI	OLA 4	SNVT_count_f	Test BO 1
●	AO 5	W+	7171	BV	2	85	NVI	OLA 5	SNVT_count_f	Test BV 2
●	AO 6	W+	7171	MO	1	85	NVI	OLA 6	SNVT_count_f	Test MO 1
●	AO 7	W+	7171	MO	2	85	NVI	OLA 7	SNVT_count_f	Test MO 2

10 Tool 'View Data' Page

10.1 Viewing Object Data

Once the gateway is fully configured and operational, you can look at object values on the View Data page. This page displays the contents of the internal data objects. In most cases, this will also represent what is found in the respective BACnet objects. If there is a one-to-one mapping of object to non-structured network variable, then this page will generally reflect what is found in the respective network variable except that NV scaling to raw data on the LonWorks network will often be different than data displayed here. What is displayed here is most often shown in human readable engineering units.

The most user friendly way to view data from the LonWorks side is to use the network management tool's NV browser, as found in Echelon's LonMaker. You can also use Nodeutil to view NV data in raw form via the LonWorks network. The third option is that you can view NV data in raw form via the USB diagnostic console (Appendix A).


LonWorks-BACnet Gateway Node Configuration Tool v2.03
Connected: ✓ Sync: ✗
Connect | Obj Import | Obj List | NV Import | NV List | Master List | View Data | BACnet Port | LonWorks
Get Object data values
Execute

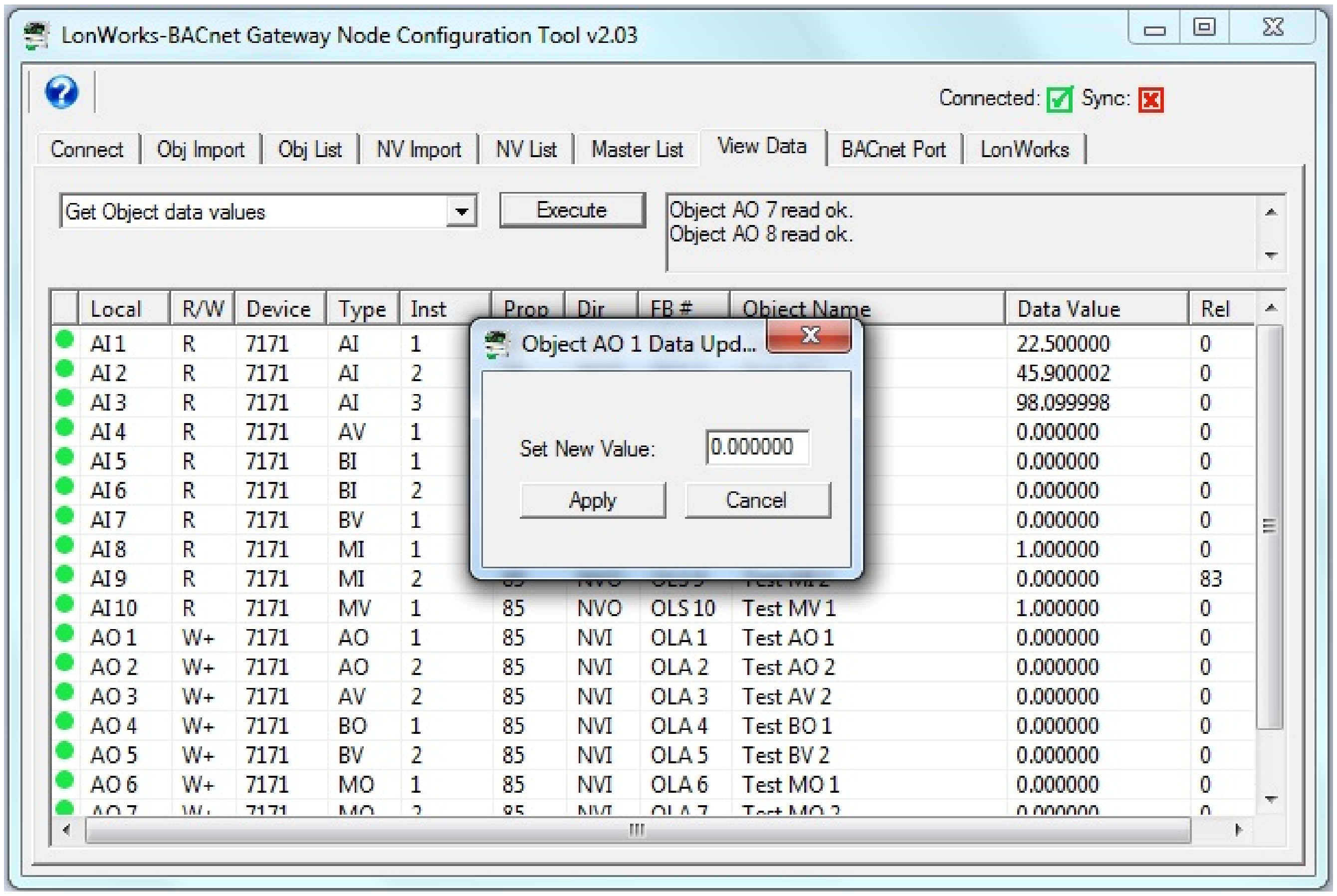
Object AO 7 read ok.
Object AO 8 read ok.

	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	Object Name	Data Value	Rel
●	AI 1	R	7171	AI	1	85	NVO	OLS 1	Test AI 1	22.500000	0
●	AI 2	R	7171	AI	2	85	NVO	OLS 2	Test AI 2	45.900002	0
●	AI 3	R	7171	AI	3	85	NVO	OLS 3	Test AI 3	98.099998	0
●	AI 4	R	7171	AV	1	85	NVO	OLS 4	Test AV 1	0.000000	0
●	AI 5	R	7171	BI	1	85	NVO	OLS 5	Test BI 1	0.000000	0
●	AI 6	R	7171	BI	2	85	NVO	OLS 6	Test BI 2	0.000000	0
●	AI 7	R	7171	BV	1	85	NVO	OLS 7	Test BV 1	0.000000	0
●	AI 8	R	7171	MI	1	85	NVO	OLS 8	Test MI 1	1.000000	0
●	AI 9	R	7171	MI	2	85	NVO	OLS 9	Test MI 2	0.000000	83
●	AI 10	R	7171	MV	1	85	NVO	OLS 10	Test MV 1	1.000000	0
●	AO 1	W+	7171	AO	1	85	NVI	OLA 1	Test AO 1	0.000000	0
●	AO 2	W+	7171	AO	2	85	NVI	OLA 2	Test AO 2	0.000000	0
●	AO 3	W+	7171	AV	2	85	NVI	OLA 3	Test AV 2	0.000000	0
●	AO 4	W+	7171	BO	1	85	NVI	OLA 4	Test BO 1	0.000000	0
●	AO 5	W+	7171	BV	2	85	NVI	OLA 5	Test BV 2	0.000000	0
●	AO 6	W+	7171	MO	1	85	NVI	OLA 6	Test MO 1	0.000000	0
●	AO 7	W+	7171	MO	2	85	NVI	OLA 7	Test MO 2	0.000000	0

10.2 Changing Object Data

Double click on any line on the View Data page to bring up the Data Update dialog. Enter a new value, and click Apply. This will write the new data value into the internal data object. Depending on your configuration, this may result in writing to a BACnet object in an external slave device, and/or may result in writing to a LonWorks Network Variable Output.

You should also be aware that, depending on your configuration, your newly set data value could get immediately overwritten. For example, if the data object is mapped to a BACnet object that is being read every 2 seconds, your value will remain in effect for only up to 2 seconds, until the next time the BACnet data is read and placed into this object.


LonWorks-BACnet Gateway Node Configuration Tool v2.03

Connected: ✓ Sync: ✗

Connect | Obj Import | Obj List | NV Import | NV List | Master List | View Data | BACnet Port | LonWorks

Get Object data values — Execute

Object AO 7 read ok.
Object AO 8 read ok.

	Local	R/W	Device	Type	Inst	Prop	Dir	FB #	Object Name	Data Value	Rel
●	AI 1	R	7171	AI	1					22.500000	0
●	AI 2	R	7171	AI	2					45.900002	0
●	AI 3	R	7171	AI	3					98.099998	0
●	AI 4	R	7171	AV	1					0.000000	0
●	AI 5	R	7171	BI	1					0.000000	0
●	AI 6	R	7171	BI	2					0.000000	0
●	AI 7	R	7171	BV	1					0.000000	0
●	AI 8	R	7171	MI	1					1.000000	0
●	AI 9	R	7171	MI	2					0.000000	83
●	AI 10	R	7171	MV	1	85	NVO	OLS 10	Test MV 1	1.000000	0
●	AO 1	W+	7171	AO	1	85	NVI	OLA 1	Test AO 1	0.000000	0
●	AO 2	W+	7171	AO	2	85	NVI	OLA 2	Test AO 2	0.000000	0
●	AO 3	W+	7171	AV	2	85	NVI	OLA 3	Test AV 2	0.000000	0
●	AO 4	W+	7171	BO	1	85	NVI	OLA 4	Test BO 1	0.000000	0
●	AO 5	W+	7171	BV	2	85	NVI	OLA 5	Test BV 2	0.000000	0
●	AO 6	W+	7171	MO	1	85	NVI	OLA 6	Test MO 1	0.000000	0
●	AO 7	W+	7171	MO	2	85	NVI	OLA 7	Test MO 2	0.000000	0

Object AO 1 Data Upd...

Set New Value: 0.000000

Apply Cancel

10.3 Diagnostics - Object Reliability

The reliability code will provide an indication of problems if non-zero. Reliability codes defined by the Babel Buster gateway are as follows:

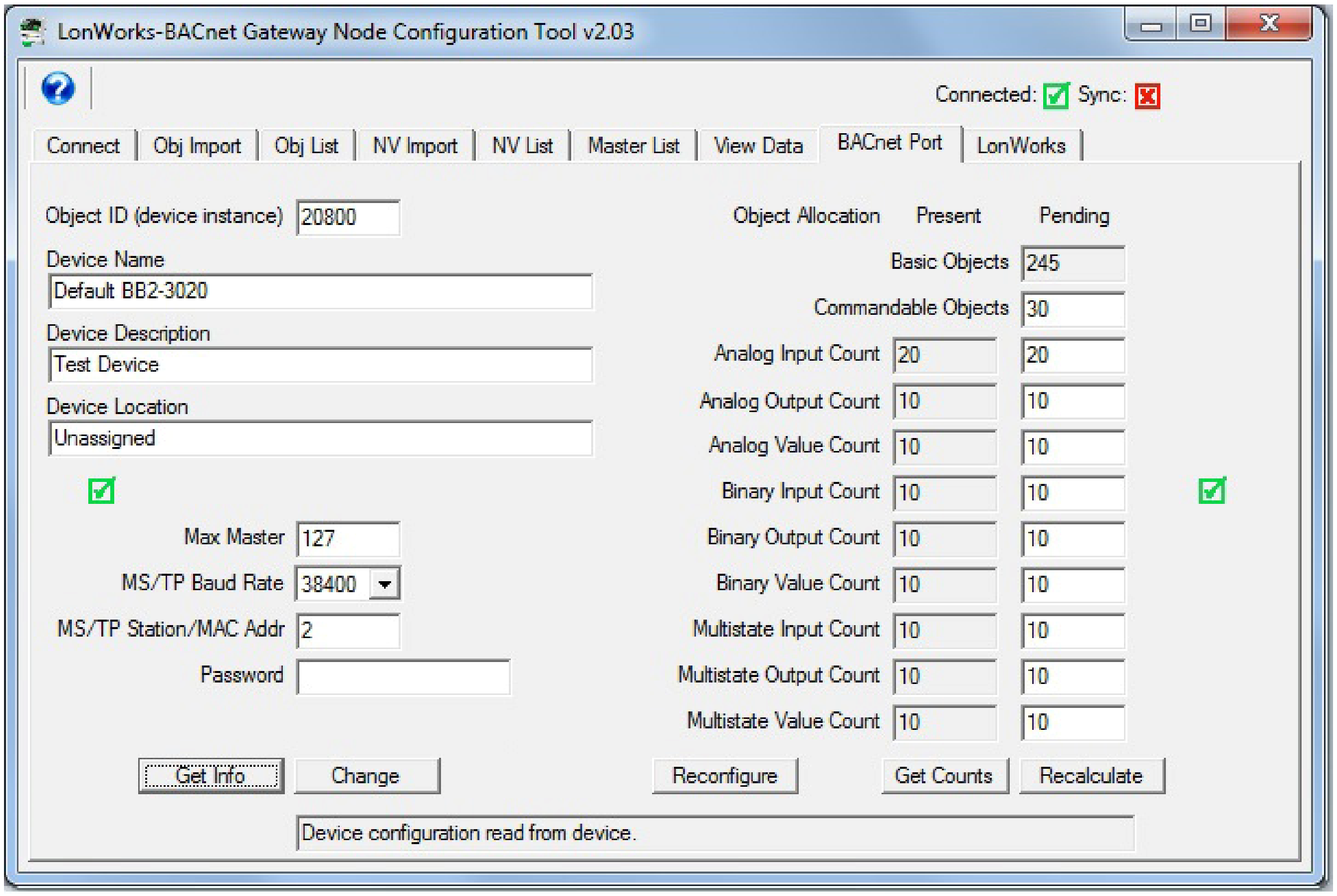
80 - Unknown or bad configuration
82 - Timeout, no response from BACnet slave/server
83 - Error code was returned by BACnet slave/server

Timeout indicates a connection problem. Error code means the device is communicating, but the request made by the Babel Buster gateway resulted in an error code being returned by the other device. Most often this means the gateway was configured to request an object instance or property that did not exist in the device being queried.

11 Tool 'BACnet Port' Page

11.1 BACnet Device Settings

The BACnet Port page is really more than port settings. It also includes device settings for representing the Babel Buster gateway as a native BACnet device on the BACnet network. Click "Get Info" to read the current device parameters. Make changes, and click "Change" to write new values to the gateway hardware.


LonWorks-BACnet Gateway Node Configuration Tool v2.03

Connected: ✓ Sync: ✗

Object ID (device instance) 20800

Device Name: Default BB2-3020

Device Description: Test Device

Device Location: Unassigned

✓

Max Master 127

MS/TP Baud Rate 38400

MS/TP Station/MAC Addr 2

Password

Object Allocation	Present	Pending
Basic Objects		245
Commandable Objects		30
Analog Input Count	20	20
Analog Output Count	10	10
Analog Value Count	10	10
Binary Input Count	10	10
Binary Output Count	10	10
Binary Value Count	10	10
Multistate Input Count	10	10
Multistate Output Count	10	10
Multistate Value Count	10	10

✓

Device configuration read from device.

Configuration parameters for the BACnet Device object are as follows:

Device Instance	Enter a device instance number that other BACnet clients will use to locate this gateway by using Who-Is and device instance.
Device Name	Enter a name for the device object. The name must be unique.
Device Description	Enter any arbitrary description that other BACnet clients may read.
Device Location	Enter any arbitrary location description that other BACnet clients may read from the device object.
Password	Enter the password that will be needed to qualify a "reinitialize device" request received from the BACnet network.

11.2 BACnet Object Counts

The BB2-3020 and BB2-7020 have a pool of up to 300 objects that may be allocated to any of the more common object types. If commandable objects are used, they will consume the pool faster due to the fact that they require more resources.

Get Counts - will read the currently configured object counts from the gateway hardware.

Recalculate - will scan the configuration currently found in the tool's Master List and set the Pending object counts to whatever is needed to accommodate that configuration.

Reconfigure - will write the Pending counts to the gateway hardware, after which Get Counts should return the same counts that were previously Pending. You can reconfigure using the calculated counts provided by Recalculate, or you can enter numbers of your own. If you anticipate adding objects, it is a good idea to configure more objects than what the Recalculate button currently says you need.

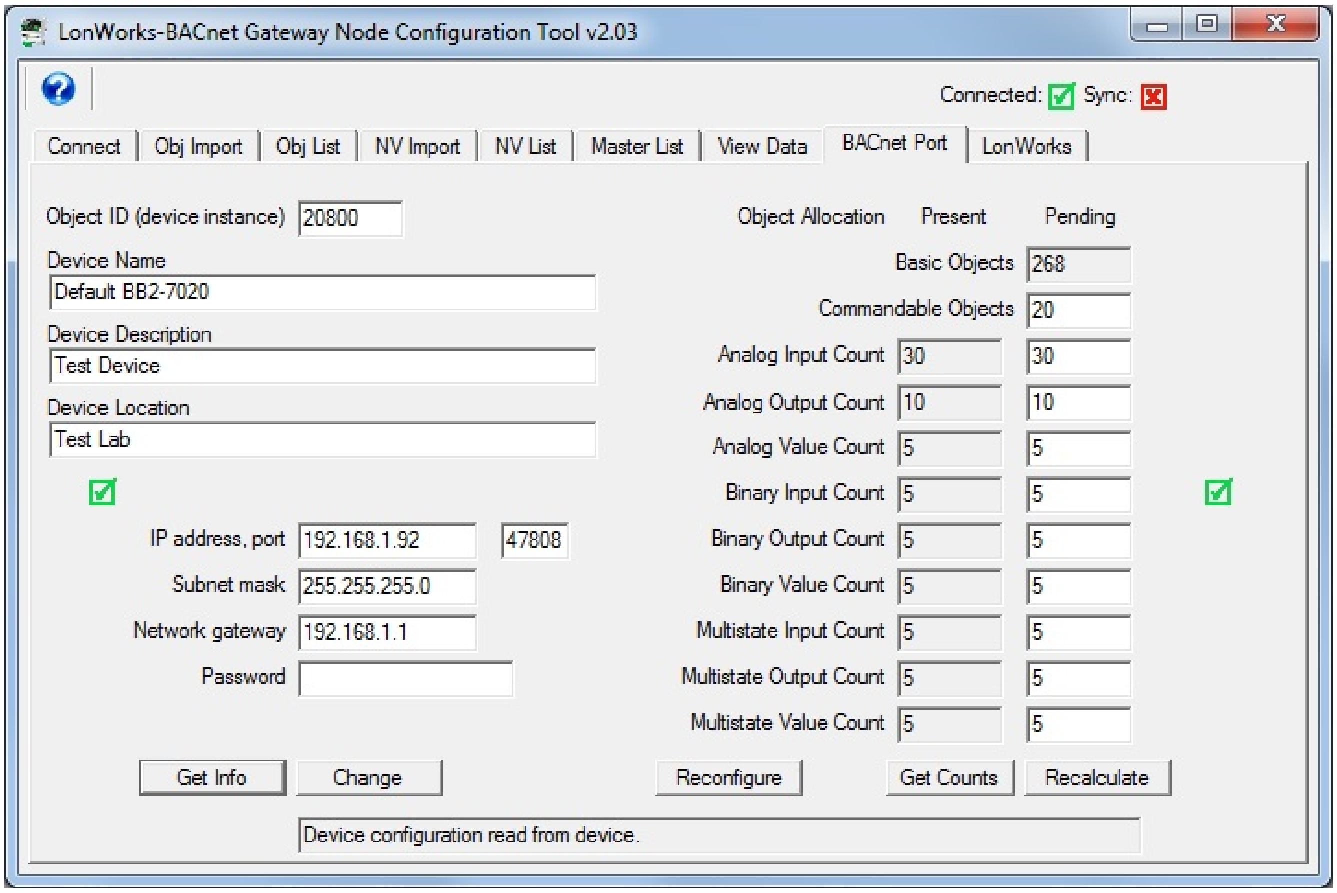
11.3 MS/TP Port Settings

The MS/TP port settings will appear if BB2-3020 has been selected as the device model on the Connect page. The port parameters that must be set for MS/TP operation are as follows. Click "Change" to write the updated parameters to the gateway hardware.

Max Master	Set this from 1 to 127. Most often, 127 is the default value used by everything else on the MS/TP network. However, if it has been changed, this setting must match in all MS/TP devices on the network in order to achieve reliable communication.
MS/TP Baud Rate	Standard MS/TP baud rates are supported. All devices on the MS/TP network must use the same baud rate.
MAC Address	MAC address is any number from 0 to 255. This is also sometimes known as the Station ID. Zero is usually a poor choice because quite often, the front end system is using zero as its MAC address. IMPORTANT: MAC address must be unique. If two or more devices on the MS/TP network have the same MAC address, you willl have serious communication problems.

11.4 IP Port Settings

The BACnet IP port settings will appear if BB2-7020 has been selected as the device model on the Connect page. The port parameters that must be set for BACnet IP operation are the standard IP address, subnet mask, and gateway common to all IP devices. Click the "Change" button to write the updated parameters to the gateway hardware. Note, however, that a changed IP address will not take effect until the next restart or power cycle of the gateway.



Object ID (device instance)	20800
Device Name	Default BB2-7020
Device Description	Test Device
Device Location	Test Lab
✓	
IP address, port	192.168.1.92 47808
Subnet mask	255.255.255.0
Network gateway	192.168.1.1
Password	

Object Allocation	Present	Pending
Basic Objects		268
Commandable Objects		20
Analog Input Count	30	30
Analog Output Count	10	10
Analog Value Count	5	5
Binary Input Count	5	5
Binary Output Count	5	5
Binary Value Count	5	5
Multistate Input Count	5	5
Multistate Output Count	5	5
Multistate Value Count	5	5

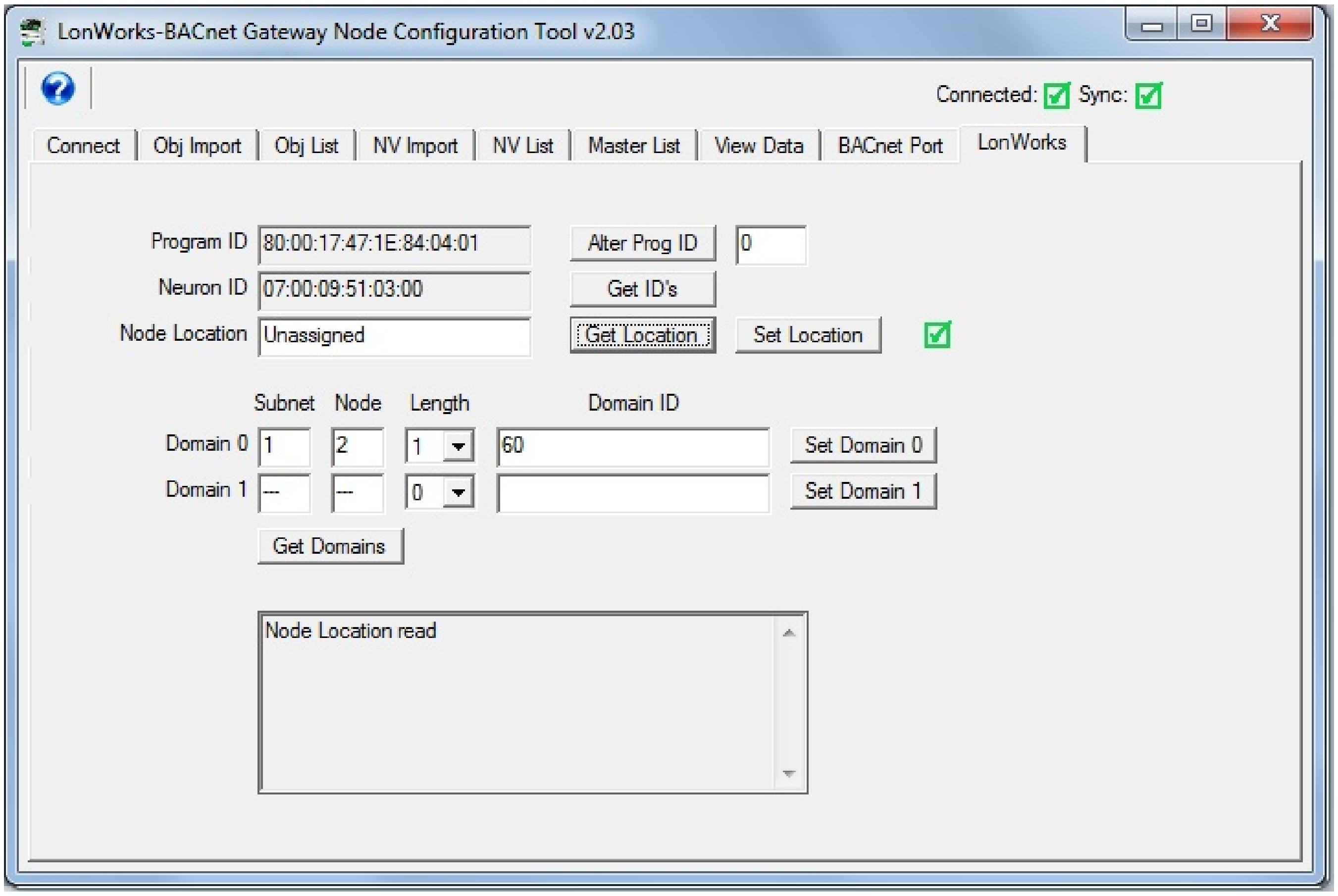
Device configuration read from device.

12 Tool 'LonWorks' Page

12.1 Viewing ID Information

Click the Get ID's button to read the Babel Buster gateway's program ID and Neuron ID. The program ID should always come back as 80:00:17:47:1E:84:04:01 where the only potential variation is that the last field, '01', may have been altered. If you get any other result, confirm that you are connected to a BB2-3020 or BB2-7020.

There is no restriction on what the Neuron ID might be. It is displayed here for reference just in case you are trying to correlate traffic in the LonScanner protocol analyzer.



Program ID: 80:00:17:47:1E:84:04:01

Alter Prog ID: 0

Neuron ID: 07:00:09:51:03:00

Node Location: Unassigned ✓

	Subnet	Node	Length	Domain ID
Domain 0	1	2	1	60
Domain 1	---	---	0	

Node Location read

12.2 Node Location

The location string provided by the device's Node Object may be read using Get Location, or written using Set Location. The location string will often get overwritten by the network management tool during network installation.

12.3 Node Domains

You have the option of viewing current domain settings using the Get Domains button, or setting the domains with the respective Set Domain buttons. In most cases, it is not necessary that you set the domains ahead of time. The network management tool will usually overwrite them anyway during installation.

Important note about being able to discover nodes on the network:

Being able to discover the gateway on the network, and the network management tool's ability to install it, depends on the device having at least one domain table entry set for a zero length domain. Without a zero length domain, the network management tool cannot query the device (in our case, the Babel Buster gateway). If your network management tool is unable to communicate with the Babel Buster, and you see here that both domains have a length greater than zero, change one of them to be zero length again.

12.4 Changing Program ID

You should normally have no reason to change the program ID. However, if you are pre-configuring multiple Babel Buster gateways with network variables that are not the same from one gateway to the next, this will look like different "program interfaces" to the network management tool. Another way of stating this is that the XIF files produced for the gateways will not be the same.

When the XIF file or program interface does not define the exact same set of network variables from one gateway to the next, you will need to create additional program ID's to allow the network management tool to see them as unique different devices. Enter a number between 21 and 255 in the ID alter window, and click Alter Prog ID. You will need to restart the gateway (power cycle) before the new ID will take effect.

Note that ID numbers 1 through 20 are reserved for Control Solutions standard gateway ID numbers. User defined program ID numbers may be 21 through 255. Note that the input window accepts a decimal number even though the complete program ID is displayed as hexadecimal.

Appendix A Diagnostics via USB Console

A.1 Connecting to Console

The USB connection to the Babel Buster gateway emulates a serial port. You can access the USB console via the Connect page. Enter commands in the command window and click Send. The result will be displayed in the log window below the command window.

If you will be using the USB console more extensively, it may be more convenient to connect via a terminal program like PuTTY (free download) or HyperTerminal (included with Windows until Win7, now you need to download it). Find the correct COM port number using your PC's device manager, and configure your terminal program to talk to that port.

Note: If the gateway is power cycled or restarted, you will most often need to disconnect USB, close the terminal program (or Control Solutions configuration software), then reconnect USB and restart your program.

A.2 Commands

Commands that you may type in using the USB console are as follows:

pr <c1 n2> - read point data for object type c1, instance n2 (and reliability code [n])
pw <c1 n2> <v> - write point data for object type c1, instance n2, value v
pcx <c1 n2 n3> - show XML point configuration for object type c1, instance n2, line n3=0..3
ps <c1 n2> - show point status for object type c1, instance n2
plist - list points - one line summary per defined point
p - alias for 'plist'

Note: c1 in the above commands is any of "ai", "ao", "av", "bi", "bo", "bv", "mi", "mo", "mv"

mdev <n> - show MS/TP device parameters, line index 'n' where n = 0..2
usage - displays message indicating counts of objects and network variables currently used.
cver - list firmware version

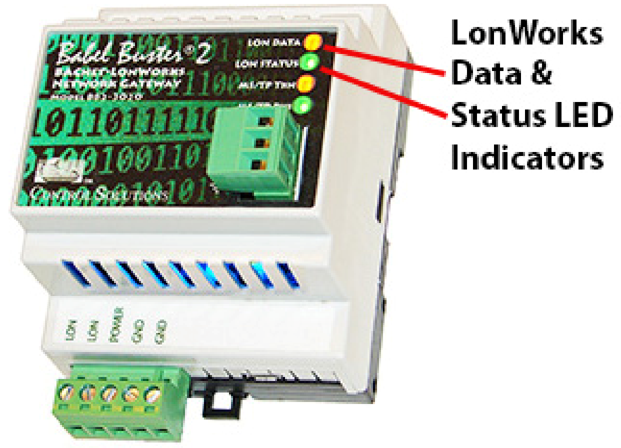
nvr <index> - read NV at 'index' (decimal), size determined automatically.
nvw <index> <b1> <b2> ... - write NV at 'index' (decimal) with hex bytes given.
Sensor NVO 1 is NV index 3, Actuator NVI 1 is NV index 123.
nvot <fbn> - displays index and SCPTnvType for NVO at sensor FB 'n', n=1..120.
nvit <fbn> - displays index and SCPTnvType for NVI at actuator FB 'n', n=1..120.
nid - displays Neuron ID
npid - displays device's Program ID
ninfo - displays node location configuration
nstate - displays state of node, single value as follows:
0: Invalid, Echelon use only
1: Invalid, Echelon use only
2: Has application, unconfigured
3: Applicationless, unconfigured
4: Configured, online (normal operating state)
5: Invalid, Echelon use only
6: Hard offline
7: Invalid, Echelon use only
0x0C: Configured, soft-offline
0x8C: Configured, in bypass mode

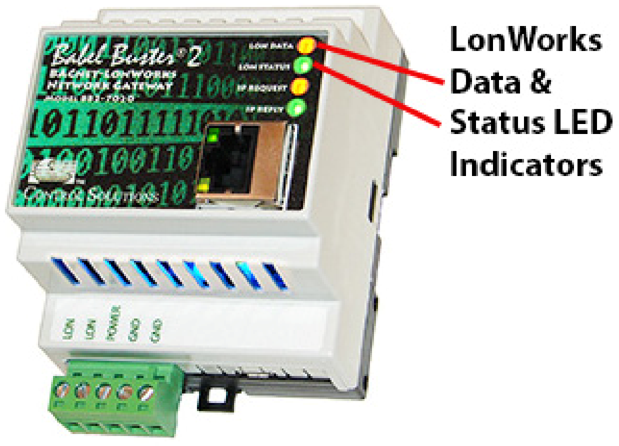
Appendix B LonWorks Trouble Shooting

B.1 General Practice, LED Indicators

The BB2-3020 and BB2-7020 are most often used to interface a BACnet device to a LonWorks network. Diagnostics favor LonWorks learning of problems with the BACnet device. Because of the peer to peer nature of LonWorks and the network variable binding, there is little the BACnet device can do to determine if there is a problem with the LonWorks device. Problems with the BACnet device are indicated to LonWorks via the Object Status.

You can exercise the LonWorks side of the gateway without commissioning on a live network by using Echelon's Nodeutil tool. This is available from the Downloads page at www.echelon.com. Using Nodeutil, you can query network variables, and also update network variable inputs in the gateway. When accessing network variable data with Nodeutil, be aware that data is provided as a series of bytes displayed in hexadecimal form. Therefore, if the variable is SNVT_temp and the data represents 25.2°C, the raw data displayed by Nodeutil will be 0B B0. If will be very useful to have a hexadecimal calculator pulled up on your PC, and have the LonMark SNVT.pdf document handy.





The LonWorks indicator LEDs display LonWorks network activity. Note, however, that activity of these LEDs is not only affected by configuration of the device, but by whether the device is commissioned on the network and whether other devices are also communicating.

Mode	Data LED	Status LED
Wink	Alternates between yellow & green 10 times, then resume normal mode.	Alternates between red & green 10 times, then resume normal mode.
Normal	Yellow flash indicates NV update was sent by gateway. Green flash indicates NV update was received by gateway.	Red indicates Neuron chip is not running. Green indicates gateway's host processor is communicating with LonWorks Neuron chip.

The "wink" behavior is invoked by sending a wink command to the BB2 gateway via the LonWorks network. This is generally just a diagnostic to see if you are successfully communicating with the device via LonWorks. Other than the few seconds it takes to execute the wink, the device will always be in "normal" mode as far as LED indications are concerned in the table above.

B.2 Object Status Bits

Object status bits that explicitly report BACnet device problems are as follows:

open_circuit

This status bit is set when the BACnet slave device (server) does not respond. The timeout is set in the client map accessible from the Master List page. If the slave does not respond within this time. the query is aborted and an error reported as open_circuit (for lack of any more appropriate definition in Object Status). This timeout also results in a BACnet object reliability code of 82.

unable_to_measure

This status bit is set when the BACnet slave device (server) returns an error code reply. This means the slave device is communicating, but the Babel Buster gateway, acting as BACnet client, has requested something that the slave is unable to provide. This typically means a configuration error. This error also results in a BACnet object reliability code of 83.

comm_failure

This status bit indicates that an internal problem was detected with the gateway configuration, or that an unknown error in communications has occurred. This error also results in a BACnet object reliability code of 80.

disabled

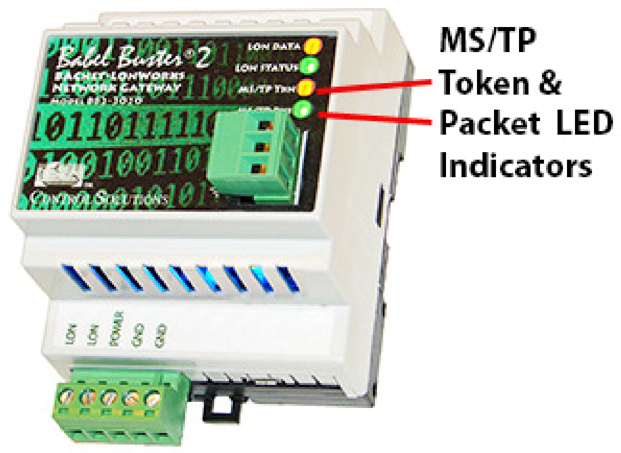
This status bit will indicate if the function block, or object, is currently disabled. Objects are expected to be disabled while configuration properties are changed via the LonWorks network, including NV type, and minimum/maximum send times. This disabling of the object is normally handled by the network management tool (e.g. LonMaker) and the object should be re-enabled by the same tool upon completion of changes.

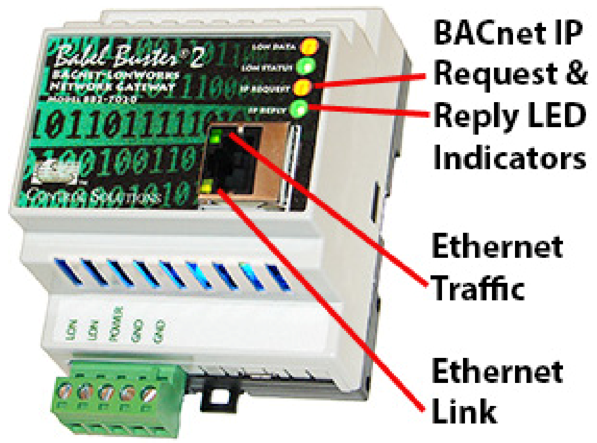
Appendix C BACnet Trouble Shooting

C.1 Observing BACnet Errors, LED Indicators

BACnet errors can be observed via Object Status bits as described in Appendix B. BACnet errors can be displayed on the View Data page. Errors are indicated by a non-zero value for object reliability ("Rel" column).

The LED indicators on the front of the gateway also indicate BACnet errors. These are global indicators that do not tell you which device or which object is having trouble, but these indicators are a very quick way to observe whether there are problems, and also whether there is any activity at all.





The MS/TP or IP LEDs will indicate BACnet traffic as indicated in the tables below.

BB2-3020 MS/TP LEDs		
Mode	MS/TP TKN (token) LED	MS/TP PKT (packet) LED
Gateway is Client or Server	Flash yellow each time the gateway sends a "poll for master". Flash green each time the token is passed.	Flash green any time a packet is sent on the network. This can be either a request or a response to a request Flash red any time an error is detected. The error can be an error code reply, or timeout. The object reliability code will indicate which one it was.

An MS/TP device that is functioning normally will always be at least passing the token, and usually polling for master periodically. The only time a device will not poll for master is if another device exists on the network with a MAC address only 1 greater than the gateway itself. It then just passes the token to that device without any intermediate polling for master.

Two particular indications of the MS/TP token LED are worth noting.

If the token LED appears to be on nearly solid yellow (or amber), this means the gateway is doing nothing but poll for token, which implies it is not finding any other devices on the network. You will see this any time you apply power to the gateway without connecting any network. If the network is connected, there is a problem with connections or port settings.

If the token LED is completely off and never flashes on, this means the gateway is hearing other traffic on the network, but just not connecting. An MS/TP device is required to "be quiet until spoken to". Therefore, if there is chatter on the network, but the gateway cannot find anything that is addressed to the gateway, it will remain quiet. This problem is usually the result of problems with port settings.

A gateway (or any MS/TP device) can alternate between the above two conditions. The "be quiet until spoken to" rule is up for grabs when the entire network is powered up simultaneously. Whoever wakes up first will be generating all the chatter whiile everybody else on the network listens.

If the token LED is very intermittent, flashes rapidly a few times (including green on the token LED), then is off completely for a period of time, this usually means it is trying to talk, thought it was talking or actually was for a bit, but something got out of sync. If there are duplicate device addresses on the network or mismatching Max Master settings, this will result in the appearance of intermittent communications.

Check you wiring. Noise on the line resulting from not following RS-485 wiring guidelines will result in communication problems ranging from no connection at all to frequent errors. Also be sure taht you have a ground connection between all devices on the network. The RS-485 network is not a 2-wire network. You need two data lines plus signal ground. If you consider cable shield a connection, then you need 4 connections.

BB2-7020 IP LEDs		
Mode	IP Request LED	IP Reply LED
Gateway is Client (Master)	Flash yellow each time gateway sends a request to a remote server (slave).	Flash green when gateway receives a good response. Flash red when gateway receives error code message from server, or if timed out with no response from server.
Gateway is Server (Slave)	Flash yellow each time gateway receives a request from external client.	Flash green when gateway recognizes request as good/valid and sends a good reply. Flash red when gateway receives a request that results in replying with an error code.

The Ethernet traffic LED will indicate any traffic on the Ethernet network, and does not necessarily indicate BACnet IP traffic. The traffic LED will typically be off more than on, flashing on each time traffic is indicated. If the traffic LED is on completely solid, the server is not running (normal for a half minute or so during startup).

The Ethernet link LED will be on any time there is a connection to the network. If the Ethernet cable is unplugged, this light will go out. If BACnet IP is failing and this light is out, check Ethernet cables.

Appendix D BACnet CSV Object List Format

D.1 Data Labels on Header Line

The required format for importing BACnet object lists into the Babel Buster LonWorks gateway from a CSV file is intended to be as forgiving as possible. The first line of the CSV file must be a header line containing labels for the columns of data.

The available column labels are outlined in the table below. The minimum set of required columns is just OBJTYPE, OBJINST. The remaining labels are optional, although in most cases at least some additional labels will be desirable. The order in which the labels appear does not matter, so long as the data on subsequent lines follows the same order as the header line. Each line of data must contain as many elements as there are in the header line. The defaults noted will apply when that label is not included in the header line.

Label	Content	What it means
RW	Either R or W (or W+) (default is R)	Enter R to read the object, or W to write it. If omitted, defaults to R. Use W+ for write on Update.
DEV	Device instance, e.g. 7171 (default is 1)	Enter the device instance of the remote BACnet server (slave). The remote device must be able to support Who-Is/I-Am for locating device by device instance.
MAC	Mac address between 0 and 127.	Sets the MS/TP Mac address (station ID) if the slave device does not support Who-Is and I-Am. Device instance will be used to automatically capture the Mac address when Who-Is/I-Am is supported. This setting should be zero to use Who-Is, or set to some non-zero Mac address if the slave does not support Who-Is. This setting applies to MS/TP only. BACnet IP devices are required to support Who-Is.
OBJTYPE	Either object type, e.g. AI, AO, etc.	Enter object type. Recognized labels are AI, AO, AV, BI, BO, BV, MI, MO, MV where "A" means Analog, "B" means Binary, "M" means Multi-state and "I" means Input, "O" means Output, "V" means Value.
OBJINST	Instance number 1 to 4194302.	Select the object instance. There are usually multiple instances of objects in a device. This identifies which one you want.
PROP	Property number, e.g. 85 (default).	Enter property code as defined by BACnet protocol, e.g. 85 is present value.
INDEX	Array index, or -1 (default) if none.	Some properties are an array of values. If applicable, enter the array index here. Enter -1 to indicate no array index is to be used. Zero is a valid index.
BITNUM	Bit number if applicable.	When the property type is a bit string, this entry may be used to specify which bit you wish to single out, especially when applying the result to a Binary object. If no bit number is provided, the entire bit string will be assembled as an integer value

		representing a mask containing all bits.

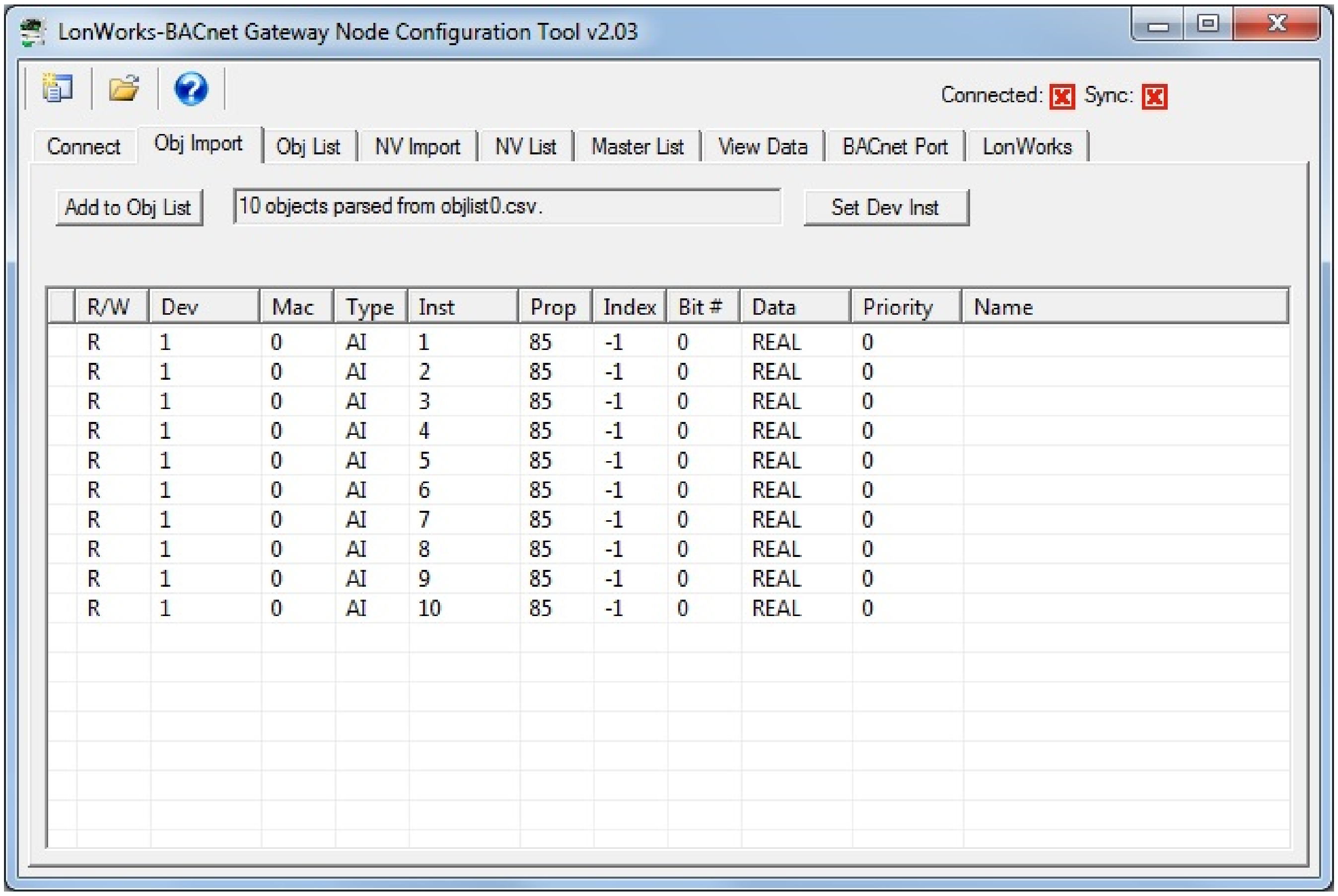
DATA	Data type NULL, BOOL, UINT, SINT, REAL, BIT, ENUM	Select the data type expected from this list. Note that some types are not supported here. Character string, for example, is one type of data that cannot be forced into Analog, Binary, or Multi-state objects by the BB2-3020 or BB2-7020.
		
		NULL = null data
		BOOL = boolean
		UINT = unsigned integer
		SINT = signed integer
		REAL = real (floating point)
		BIT = bit string
		ENUM = enumeration
		
		IMPORTANT: While you typically may think of a Binary object as boolean (or even just a bit), BACnet protocol says the proper data type for the Binary object is ENUMERATION (ENUM).

PRIORITY	Priority 1-16 if applicable.	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.
UNITS	BACnet engineering units code.	Enter BACnet protocol code for units, for example 95 is "no units" (default), "degrees Celcius" is 62, etc.
NAME	Any unique character string up to 40 characters	Enter an alphanumeric name for the object. The name must be unique, and duplicate names will result in an error.
DESC	Any arbitrary character string up to 40 characters	Enter any arbitrary description for the object. The description is not required to be unique.

D.2 Example CSV Files and Imports

The following are examples of rather simple CSV files with BACnet objects and a screen shot of the resulting import. While brief, these examples are intended to show some of the possible variations in format.

```
OBJTYPE,OBJINST
AI,1
AI,2
AI,3
AI,4
AI,5
AI,6
AI,7
AI,8
AI,9
AI,10
```
	The simplest CSV file for object import is just a list of types and instances. Device instance will default to 1, property 85 (present value).
	
	Importing this CSV results in the following screen shot.



10 objects parsed from objlist0.csv.

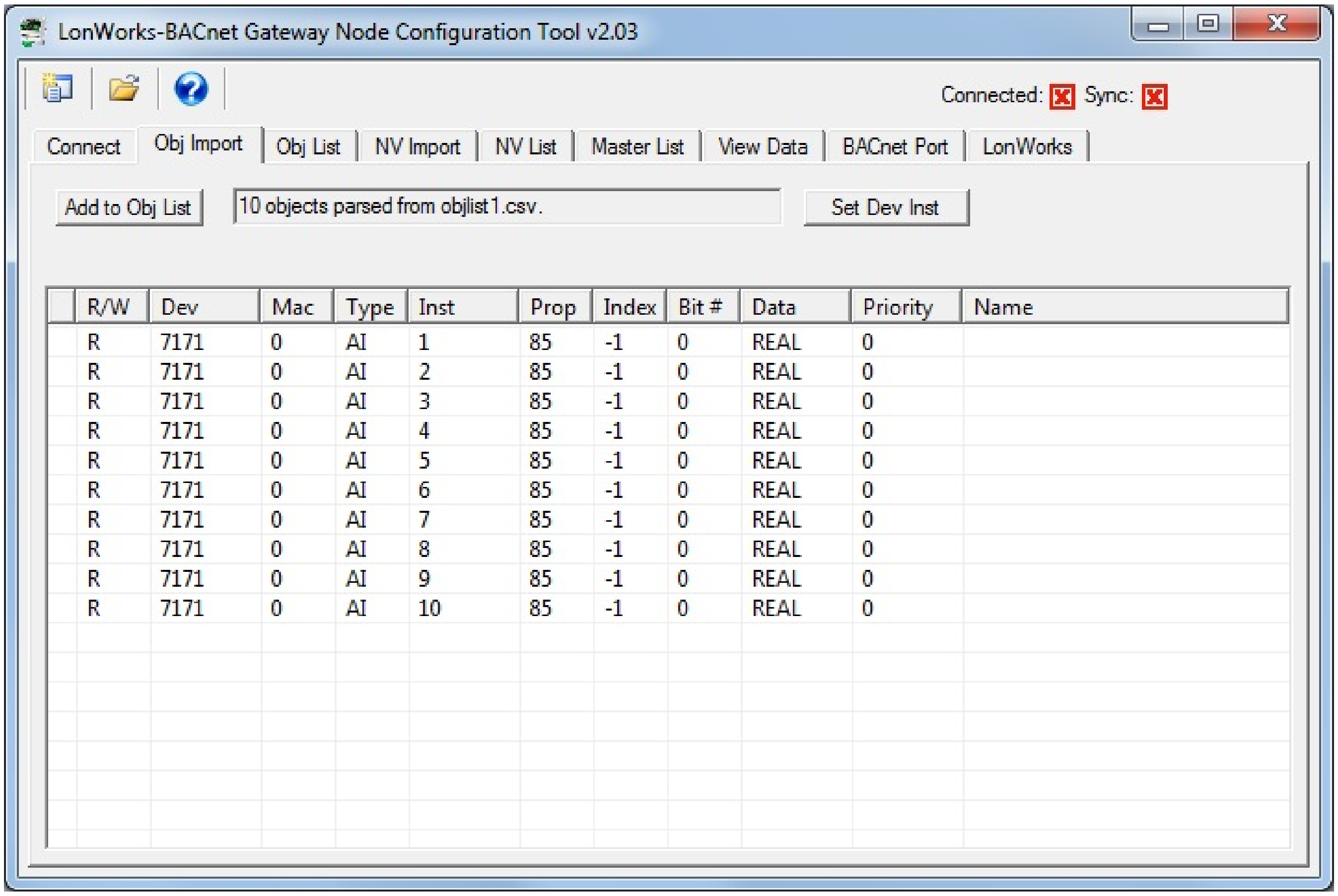
R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
R	1	0	AI	1	85	-1	0	REAL	0	
R	1	0	AI	2	85	-1	0	REAL	0	
R	1	0	AI	3	85	-1	0	REAL	0	
R	1	0	AI	4	85	-1	0	REAL	0	
R	1	0	AI	5	85	-1	0	REAL	0	
R	1	0	AI	6	85	-1	0	REAL	0	
R	1	0	AI	7	85	-1	0	REAL	0	
R	1	0	AI	8	85	-1	0	REAL	0	
R	1	0	AI	9	85	-1	0	REAL	0	
R	1	0	AI	10	85	-1	0	REAL	0	

<table>
<tr><td>

```
DEV,OBJTYPE,OBJINST
7171,AI,1
7171,AI,2
7171,AI,3
7171,AI,4
7171,AI,5
7171,AI,6
7171,AI,7
7171,AI,8
7171,AI,9
7171,AI,10
```

</td><td>A minor expansion of the above example adds the device instance column.

The following screen shot is the result of importing this CSV file.</td></tr>
</table>



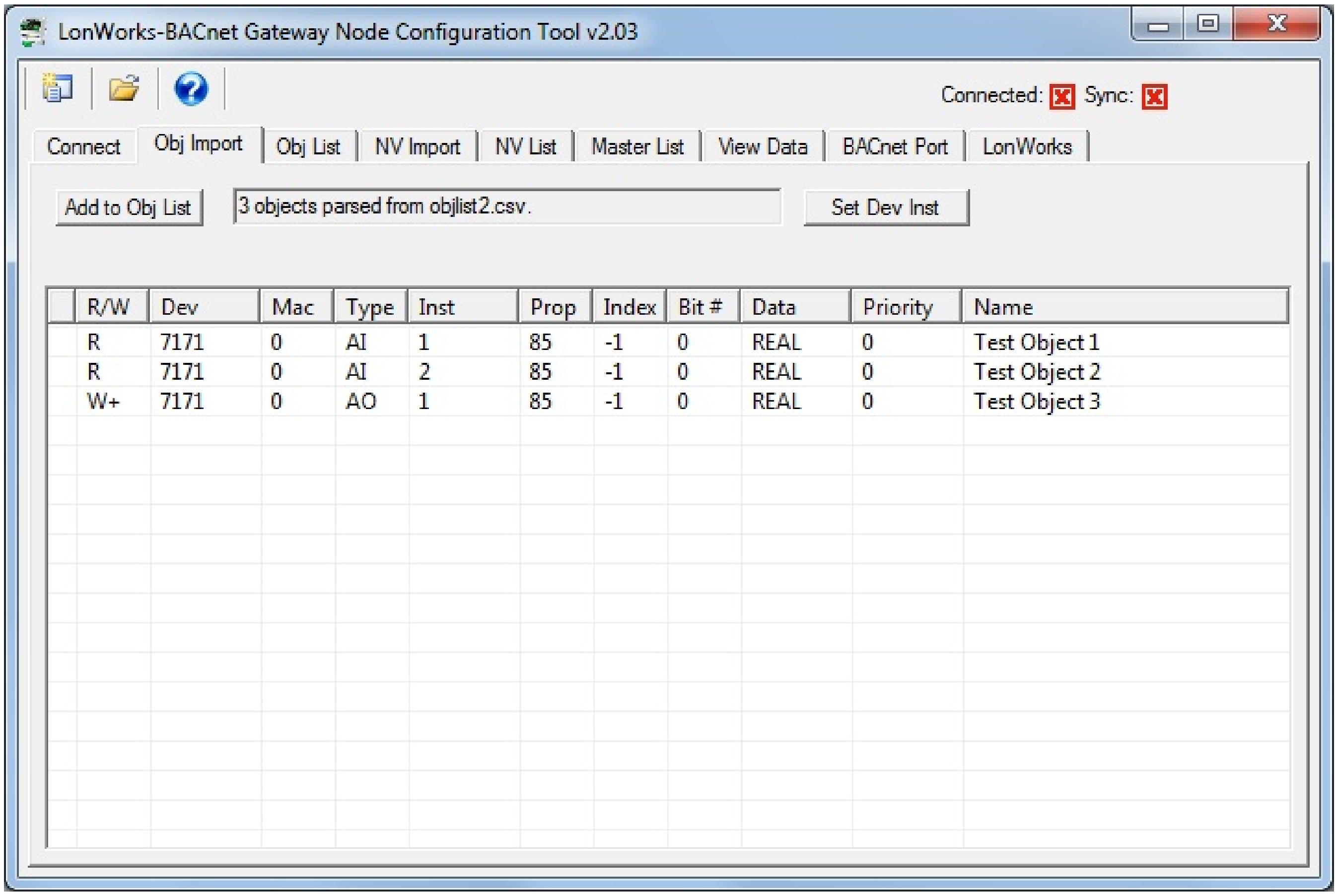
10 objects parsed from objlist1.csv.

	R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
	R	7171	0	AI	1	85	-1	0	REAL	0	
	R	7171	0	AI	2	85	-1	0	REAL	0	
	R	7171	0	AI	3	85	-1	0	REAL	0	
	R	7171	0	AI	4	85	-1	0	REAL	0	
	R	7171	0	AI	5	85	-1	0	REAL	0	
	R	7171	0	AI	6	85	-1	0	REAL	0	
	R	7171	0	AI	7	85	-1	0	REAL	0	
	R	7171	0	AI	8	85	-1	0	REAL	0	
	R	7171	0	AI	9	85	-1	0	REAL	0	
	R	7171	0	AI	10	85	-1	0	REAL	0	

```
RW,DEV,OBJTYPE,OBJINST,PROP,DATA,NAME
R,7171,AI,1,85,REAL,Test Object 1
R,7171,AI,2,85,REAL,Test Object 2
W+,7171,AO,1,85,REAL,Test Object 3
```

An example showing additional columns, including RW - in this case we are both reading and writing. Without the RW column, all lines default to Read.

Importing this CSV results in the following screen shot.



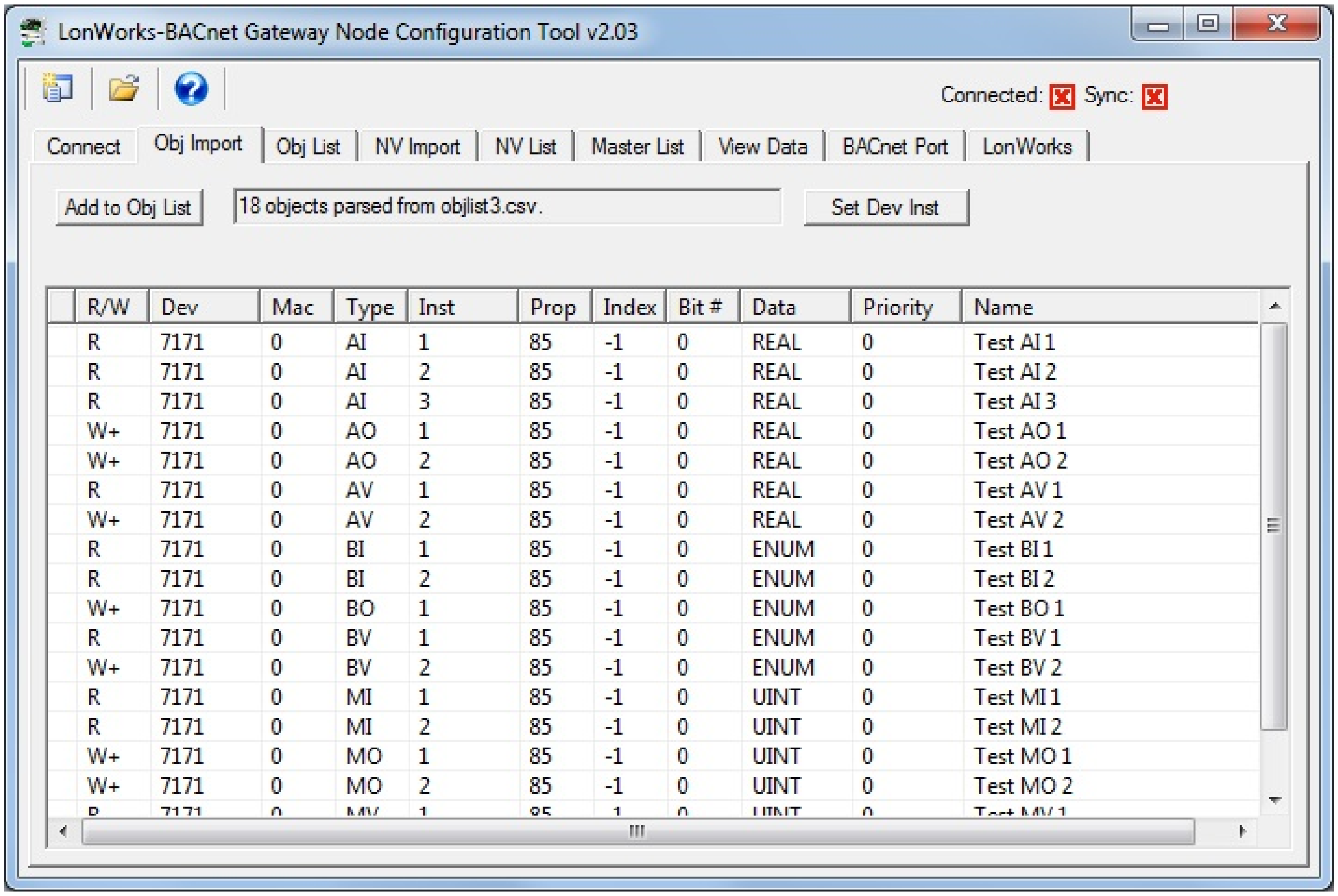
3 objects parsed from objlist2.csv.

	R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
	R	7171	0	AI	1	85	-1	0	REAL	0	Test Object 1
	R	7171	0	AI	2	85	-1	0	REAL	0	Test Object 2
	W+	7171	0	AO	1	85	-1	0	REAL	0	Test Object 3

```
RW,DEV,OBJTYPE,OBJINST,PROP,DATA,NAME
R,7171,AI,1,85,REAL,Test AI 1
R,7171,AI,2,85,REAL,Test AI 2
R,7171,AI,3,85,REAL,Test AI 3
W+,7171,AO,1,85,REAL,Test AO 1
W+,7171,AO,2,85,REAL,Test AO 2
R,7171,AV,1,85,REAL,Test AV 1
W+,7171,AV,2,85,REAL,Test AV 2
R,7171,BI,1,85,ENUM,Test BI 1
R,7171,BI,2,85,ENUM,Test BI 2
W+,7171,BO,1,85,ENUM,Test BO 1
R,7171,BV,1,85,ENUM,Test BV 1
W+,7171,BV,2,85,ENUM,Test BV 2
R,7171,MI,1,85,UINT,Test MI 1
R,7171,MI,2,85,UINT,Test MI 2
W+,7171,MO,1,85,UINT,Test MO 1
W+,7171,MO,2,85,UINT,Test MO 2
R,7171,MV,1,85,UINT,Test MV 1
W+,7171,MV,2,85,UINT,Test MV 2
```

An expanded example showing multiple object types, reading, and writing upon update (W+).

Importing this CSV results in the following screen shot.



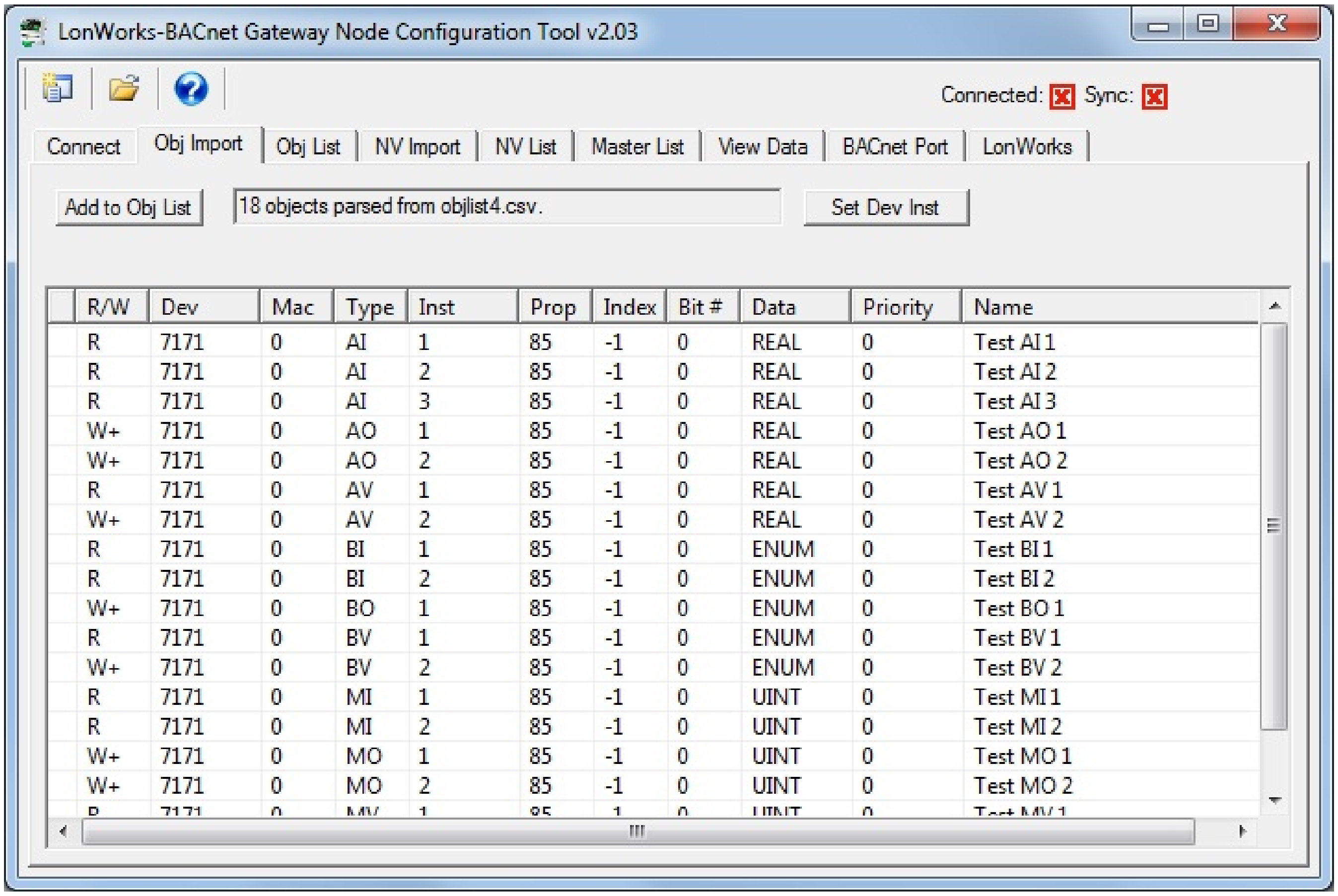
18 objects parsed from objlist3.csv.

R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
R	7171	0	AI	1	85	-1	0	REAL	0	Test AI 1
R	7171	0	AI	2	85	-1	0	REAL	0	Test AI 2
R	7171	0	AI	3	85	-1	0	REAL	0	Test AI 3
W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO 1
W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO 2
R	7171	0	AV	1	85	-1	0	REAL	0	Test AV 1
W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV 2
R	7171	0	BI	1	85	-1	0	ENUM	0	Test BI 1
R	7171	0	BI	2	85	-1	0	ENUM	0	Test BI 2
W+	7171	0	BO	1	85	-1	0	ENUM	0	Test BO 1
R	7171	0	BV	1	85	-1	0	ENUM	0	Test BV 1
W+	7171	0	BV	2	85	-1	0	ENUM	0	Test BV 2
R	7171	0	MI	1	85	-1	0	UINT	0	Test MI 1
R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1

```
RW,DEV,OBJTYPE,OBJINST,PROP,DATA,NAME,UNITS,DESC
R,7171,AI,1,85,REAL,Test AI 1,62,Description 1
R,7171,AI,2,85,REAL,Test AI 2,62,Description 2
R,7171,AI,3,85,REAL,Test AI 3,62,Description 3
W+,7171,AO,1,85,REAL,Test AO 1,95,Description 4
W+,7171,AO,2,85,REAL,Test AO 2,95,Description 5
R,7171,AV,1,85,REAL,Test AV 1,95,Description 6
W+,7171,AV,2,85,REAL,Test AV 2,95,Description 7
R,7171,BI,1,85,ENUM,Test BI 1,95,Description 8
R,7171,BI,2,85,ENUM,Test BI 2,95,Description 9
W+,7171,BO,1,85,ENUM,Test BO 1,95,Description 10
R,7171,BV,1,85,ENUM,Test BV 1,95,Description 11
W+,7171,BV,2,85,ENUM,Test BV 2,95,Description 12
R,7171,MI,1,85,UINT,Test MI 1,95,Description 13
R,7171,MI,2,85,UINT,Test MI 2,95,Description 14
W+,7171,MO,1,85,UINT,Test MO 1,95,Description 15
W+,7171,MO,2,85,UINT,Test MO 2,95,Description 16
R,7171,MV,1,85,UINT,Test MV 1,95,Description 17
W+,7171,MV,2,85,UINT,Test MV 2,95,Description 18
```

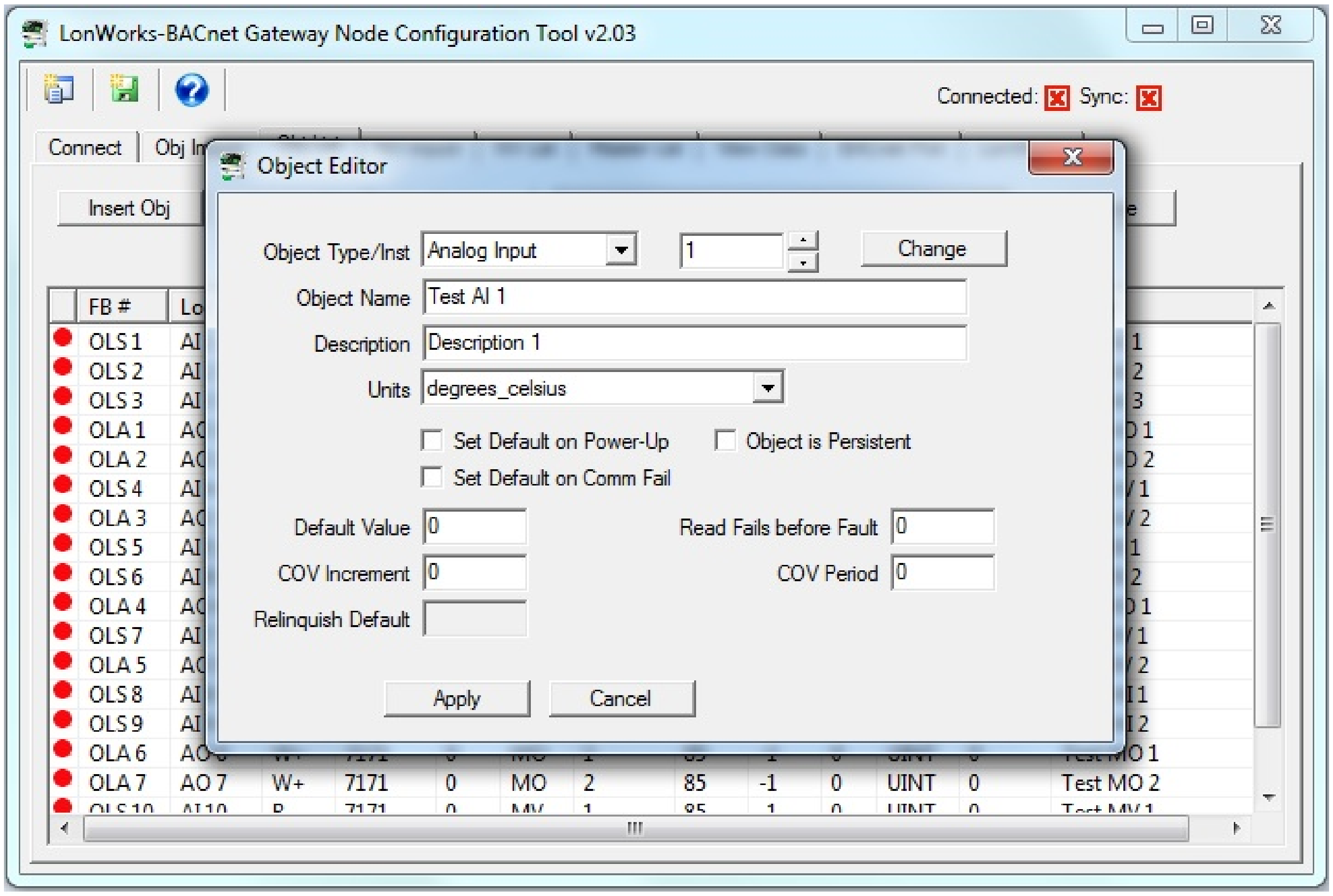
This example is the same as above with the addition of units, and description. Note that the units and description do not show up on the import list, but when incorporated into the Object List, these values do get copied into the local object as illustrated in the second screen shot below.

Importing this CSV results in the following screen shot.



18 objects parsed from objlist4.csv.

R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
R	7171	0	AI	1	85	-1	0	REAL	0	Test AI 1
R	7171	0	AI	2	85	-1	0	REAL	0	Test AI 2
R	7171	0	AI	3	85	-1	0	REAL	0	Test AI 3
W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO 1
W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO 2
R	7171	0	AV	1	85	-1	0	REAL	0	Test AV 1
W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV 2
R	7171	0	BI	1	85	-1	0	ENUM	0	Test BI 1
R	7171	0	BI	2	85	-1	0	ENUM	0	Test BI 2
W+	7171	0	BO	1	85	-1	0	ENUM	0	Test BO 1
R	7171	0	BV	1	85	-1	0	ENUM	0	Test BV 1
W+	7171	0	BV	2	85	-1	0	ENUM	0	Test BV 2
R	7171	0	MI	1	85	-1	0	UINT	0	Test MI 1
R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO 1
W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
R	7171	0	MV	1	85	-1	0	UINT	0	Test MV 1



Object Type/Inst: Analog Input, 1

Object Name: Test AI 1

Description: Description 1

Units: degrees_celsius

☐ Set Default on Power-Up ☐ Object is Persistent

☐ Set Default on Comm Fail

Default Value: 0 Read Fails before Fault: 0

COV Increment: 0 COV Period: 0

Relinquish Default:

OLA 7	AO 7	W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2

Appendix E BACnet Codes

E.1 BACnet Object Property Codes

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

```
typedef enum {
    PROP_ACKED_TRANSITIONS = 0,
    PROP_ACK_REQUIRED = 1,
    PROP_ACTION = 2,
    PROP_ACTION_TEXT = 3,
    PROP_ACTIVE_TEXT = 4,
    PROP_ACTIVE_VT_SESSIONS = 5,
    PROP_ALARM_VALUE = 6,
    PROP_ALARM_VALUES = 7,
    PROP_ALL = 8,
    PROP_ALL_WRITES_SUCCESSFUL = 9,
    PROP_APDU_SEGMENT_TIMEOUT = 10,
    PROP_APDU_TIMEOUT = 11,
    PROP_APPLICATION_SOFTWARE_VERSION = 12,
    PROP_ARCHIVE = 13,
    PROP_BIAS = 14,
    PROP_CHANGE_OF_STATE_COUNT = 15,
    PROP_CHANGE_OF_STATE_TIME = 16,
    PROP_NOTIFICATION_CLASS = 17,
    PROP_BLANK_1 = 18,
    PROP_CONTROLLED_VARIABLE_REFERENCE = 19,
    PROP_CONTROLLED_VARIABLE_UNITS = 20,
    PROP_CONTROLLED_VARIABLE_VALUE = 21,
    PROP_COV_INCREMENT = 22,
    PROP_DATE_LIST = 23,
    PROP_DAYLIGHT_SAVINGS_STATUS = 24,
    PROP_DEADBAND = 25,
    PROP_DERIVATIVE_CONSTANT = 26,
    PROP_DERIVATIVE_CONSTANT_UNITS = 27,
    PROP_DESCRIPTION = 28,
    PROP_DESCRIPTION_OF_HALT = 29,
    PROP_DEVICE_ADDRESS_BINDING = 30,
    PROP_DEVICE_TYPE = 31,
    PROP_EFFECTIVE_PERIOD = 32,
    PROP_ELAPSED_ACTIVE_TIME = 33,
    PROP_ERROR_LIMIT = 34,
    PROP_EVENT_ENABLE = 35,
    PROP_EVENT_STATE = 36,
    PROP_EVENT_TYPE = 37,
    PROP_EXCEPTION_SCHEDULE = 38,
    PROP_FAULT_VALUES = 39,
    PROP_FEEDBACK_VALUE = 40,
    PROP_FILE_ACCESS_METHOD = 41,
    PROP_FILE_SIZE = 42,
    PROP_FILE_TYPE = 43,
    PROP_FIRMWARE_REVISION = 44,
    PROP_HIGH_LIMIT = 45,
    PROP_INACTIVE_TEXT = 46,
    PROP_IN_PROCESS = 47,
    PROP_INSTANCE_OF = 48,
    PROP_INTEGRAL_CONSTANT = 49,
    PROP_INTEGRAL_CONSTANT_UNITS = 50,
    PROP_ISSUE_CONFIRMED_NOTIFICATIONS = 51,
    PROP_LIMIT_ENABLE = 52,
    PROP_LIST_OF_GROUP_MEMBERS = 53,
    PROP_LIST_OF_OBJECT_PROPERTY_REFERENCES = 54,
    PROP_LIST_OF_SESSION_KEYS = 55,
    PROP_LOCAL_DATE = 56,
    PROP_LOCAL_TIME = 57,
    PROP_LOCATION = 58,
    PROP_LOW_LIMIT = 59,
    PROP_MANIPULATED_VARIABLE_REFERENCE = 60,
    PROP_MAXIMUM_OUTPUT = 61,
    PROP_MAX_APDU_LENGTH_ACCEPTED = 62,
    PROP_MAX_INFO_FRAMES = 63,
    PROP_MAX_MASTER = 64,
    PROP_MAX_PRES_VALUE = 65,
    PROP_MINIMUM_OFF_TIME = 66,
    PROP_MINIMUM_ON_TIME = 67,
    PROP_MINIMUM_OUTPUT = 68,
    PROP_MIN_PRES_VALUE = 69,
    PROP_MODEL_NAME = 70,
    PROP_MODIFICATION_DATE = 71,
    PROP_NOTIFY_TYPE = 72,
    PROP_NUMBER_OF_APDU_RETRIES = 73,
    PROP_NUMBER_OF_STATES = 74,
    PROP_OBJECT_IDENTIFIER = 75,
    PROP_OBJECT_LIST = 76,
    PROP_OBJECT_NAME = 77,
    PROP_OBJECT_PROPERTY_REFERENCE = 78,
    PROP_OBJECT_TYPE = 79,
    PROP_OPTIONAL = 80,
    PROP_OUT_OF_SERVICE = 81,
    PROP_OUTPUT_UNITS = 82,
    PROP_EVENT_PARAMETERS = 83,
    PROP_POLARITY = 84,
    PROP_PRESENT_VALUE = 85,
    PROP_PRIORITY = 86,
    PROP_PRIORITY_ARRAY = 87,
    PROP_PRIORITY_FOR_WRITING = 88,
    PROP_PROCESS_IDENTIFIER = 89,
    PROP_PROGRAM_CHANGE = 90,
    PROP_PROGRAM_LOCATION = 91,
    PROP_PROGRAM_STATE = 92,
    PROP_PROPORTIONAL_CONSTANT = 93,
    PROP_PROPORTIONAL_CONSTANT_UNITS = 94,
    PROP_PROTOCOL_CONFORMANCE_CLASS = 95,       /* deleted in version 1 revision 2
*/
    PROP_PROTOCOL_OBJECT_TYPES_SUPPORTED = 96,
    PROP_PROTOCOL_SERVICES_SUPPORTED = 97,
    PROP_PROTOCOL_VERSION = 98,
    PROP_READ_ONLY = 99,
    PROP_REASON_FOR_HALT = 100,
    PROP_RECIPIENT = 101,
    PROP_RECIPIENT_LIST = 102,
    PROP_RELIABILITY = 103,
    PROP_RELINQUISH_DEFAULT = 104,
    PROP_REQUIRED = 105,
    PROP_RESOLUTION = 106,
    PROP_SEGMENTATION_SUPPORTED = 107,
    PROP_SETPOINT = 108,
    PROP_SETPOINT_REFERENCE = 109,
    PROP_STATE_TEXT = 110,
    PROP_STATUS_FLAGS = 111,
    PROP_SYSTEM_STATUS = 112,
    PROP_TIME_DELAY = 113,
    PROP_TIME_OF_ACTIVE_TIME_RESET = 114,
    PROP_TIME_OF_STATE_COUNT_RESET = 115,
    PROP_TIME_SYNCHRONIZATION_RECIPIENTS = 116,
    PROP_UNITS = 117,
    PROP_UPDATE_INTERVAL = 118,
    PROP_UTC_OFFSET = 119,
    PROP_VENDOR_IDENTIFIER = 120,
    PROP_VENDOR_NAME = 121,
    PROP_VT_CLASSES_SUPPORTED = 122,
    PROP_WEEKLY_SCHEDULE = 123,
    PROP_ATTEMPTED_SAMPLES = 124,
    PROP_AVERAGE_VALUE = 125,
    PROP_BUFFER_SIZE = 126,
    PROP_CLIENT_COV_INCREMENT = 127,
    PROP_COV_RESUBSCRIPTION_INTERVAL = 128,
    PROP_CURRENT_NOTIFY_TIME = 129,
    PROP_EVENT_TIME_STAMPS = 130,
    PROP_LOG_BUFFER = 131,
    PROP_LOG_DEVICE_OBJECT = 132,
    /* The enable property is renamed from log-enable in
       Addendum b to ANSI/ASHRAE 135-2004(135b-2) */
    PROP_ENABLE = 133,
    PROP_LOG_INTERVAL = 134,
    PROP_MAXIMUM_VALUE = 135,
    PROP_MINIMUM_VALUE = 136,
    PROP_NOTIFICATION_THRESHOLD = 137,
    PROP_PREVIOUS_NOTIFY_TIME = 138,
    PROP_PROTOCOL_REVISION = 139,
    PROP_RECORDS_SINCE_NOTIFICATION = 140,
    PROP_RECORD_COUNT = 141,
    PROP_START_TIME = 142,
    PROP_STOP_TIME = 143,
    PROP_STOP_WHEN_FULL = 144,
    PROP_TOTAL_RECORD_COUNT = 145,
    PROP_VALID_SAMPLES = 146,
    PROP_WINDOW_INTERVAL = 147,
    PROP_WINDOW_SAMPLES = 148,
    PROP_MAXIMUM_VALUE_TIMESTAMP = 149,
    PROP_MINIMUM_VALUE_TIMESTAMP = 150,
    PROP_VARIANCE_VALUE = 151,
    PROP_ACTIVE_COV_SUBSCRIPTIONS = 152,
    PROP_BACKUP_FAILURE_TIMEOUT = 153,
    PROP_CONFIGURATION_FILES = 154,
    PROP_DATABASE_REVISION = 155,
    PROP_DIRECT_READING = 156,
    PROP_LAST_RESTORE_TIME = 157,
    PROP_MAINTENANCE_REQUIRED = 158,
    PROP_MEMBER_OF = 159,
    PROP_MODE = 160,
    PROP_OPERATION_EXPECTED = 161,
    PROP_SETTING = 162,
    PROP_SILENCED = 163,
    PROP_TRACKING_VALUE = 164,
    PROP_ZONE_MEMBERS = 165,
    PROP_LIFE_SAFETY_ALARM_VALUES = 166,
    PROP_MAX_SEGMENTS_ACCEPTED = 167,
    PROP_PROFILE_NAME = 168,
    PROP_AUTO_SLAVE_DISCOVERY = 169,
    PROP_MANUAL_SLAVE_ADDRESS_BINDING = 170,
    PROP_SLAVE_ADDRESS_BINDING = 171,
    PROP_SLAVE_PROXY_ENABLE = 172,
    PROP_LAST_NOTIFY_TIME = 173,
    PROP_SCHEDULE_DEFAULT = 174,
    PROP_ACCEPTED_MODES = 175,
    PROP_ADJUST_VALUE = 176,
    PROP_COUNT = 177,
    PROP_COUNT_BEFORE_CHANGE = 178,
    PROP_COUNT_CHANGE_TIME = 179,
    PROP_COV_PERIOD = 180,
    PROP_INPUT_REFERENCE = 181,
    PROP_LIMIT_MONITORING_INTERVAL = 182,
    PROP_LOGGING_DEVICE = 183,
    PROP_LOGGING_RECORD = 184,
    PROP_PRESCALE = 185,
    PROP_PULSE_RATE = 186,
    PROP_SCALE = 187,
    PROP_SCALE_FACTOR = 188,
    PROP_UPDATE_TIME = 189,
    PROP_VALUE_BEFORE_CHANGE = 190,
    PROP_VALUE_SET = 191,
    PROP_VALUE_CHANGE_TIME = 192,
    /* enumerations 193-206 are new */
    PROP_ALIGN_INTERVALS = 193,
    PROP_GROUP_MEMBER_NAMES = 194,
    PROP_INTERVAL_OFFSET = 195,
    PROP_LAST_RESTART_REASON = 196,
    PROP_LOGGING_TYPE = 197,
    PROP_MEMBER_STATUS_FLAGS = 198,
    PROP_NOTIFICATION_PERIOD = 199,
    PROP_PREVIOUS_NOTIFY_RECORD = 200,
    PROP_REQUESTED_UPDATE_INTERVAL = 201,
    PROP_RESTART_NOTIFICATION_RECIPIENTS = 202,
    PROP_TIME_OF_DEVICE_RESTART = 203,
    PROP_TIME_SYNCHRONIZATION_INTERVAL = 204,
    PROP_TRIGGER = 205,
    PROP_UTC_TIME_SYNCHRONIZATION_RECIPIENTS = 206,
    /* enumerations 207-211 are used in Addendum d to ANSI/ASHRAE 135-2004 */
    PROP_NODE_SUBTYPE = 207,
    PROP_NODE_TYPE = 208,
    PROP_STRUCTURED_OBJECT_LIST = 209,
    PROP_SUBORDINATE_ANNOTATIONS = 210,
    PROP_SUBORDINATE_LIST = 211,
    /* enumerations 212-225 are used in Addendum e to ANSI/ASHRAE 135-2004 */
    PROP_ACTUAL_SHED_LEVEL = 212,
    PROP_DUTY_WINDOW = 213,
    PROP_EXPECTED_SHED_LEVEL = 214,
    PROP_FULL_DUTY_BASELINE = 215,
    /* enumerations 216-217 are used in Addendum i to ANSI/ASHRAE 135-2004 */
    PROP_BLINK_PRIORITY_THRESHOLD = 216,
    PROP_BLINK_TIME = 217,
    /* enumerations 212-225 are used in Addendum e to ANSI/ASHRAE 135-2004 */
    PROP_REQUESTED_SHED_LEVEL = 218,
    PROP_SHED_DURATION = 219,
    PROP_SHED_LEVEL_DESCRIPTIONS = 220,
    PROP_SHED_LEVELS = 221,
    PROP_STATE_DESCRIPTION = 222,
    /* enumerations 223-225 are used in Addendum i to ANSI/ASHRAE 135-2004 */
    PROP_FADE_TIME = 223,
    PROP_LIGHTING_COMMAND = 224,
    PROP_LIGHTING_COMMAND_PRIORITY = 225,
    /* enumerations 226-235 are used in Addendum f to ANSI/ASHRAE 135-2004 */
    PROP_DOOR_ALARM_STATE = 226,
    PROP_DOOR_EXTENDED_PULSE_TIME = 227,
    PROP_DOOR_MEMBERS = 228,
    PROP_DOOR_OPEN_TOO_LONG_TIME = 229,
    PROP_DOOR_PULSE_TIME = 230,
    PROP_DOOR_STATUS = 231,
    PROP_DOOR_UNLOCK_DELAY_TIME = 232,
    PROP_LOCK_STATUS = 233,
    PROP_MASKED_ALARM_VALUES = 234,
    PROP_SECURED_STATUS = 235,
    /* enumerations 236-243 are used in Addendum i to ANSI/ASHRAE 135-2004 */
    PROP_OFF_DELAY = 236,
    PROP_ON_DELAY = 237,
    PROP_POWER = 238,
    PROP_POWER_ON_VALUE = 239,
    PROP_PROGRESS_VALUE = 240,
    PROP_RAMP_RATE = 241,
    PROP_STEP_INCREMENT = 242,
    PROP_SYSTEM_FAILURE_VALUE = 243,
    /* enumerations 244-311 are used in Addendum j to ANSI/ASHRAE 135-2004 */
    PROP_ABSENTEE_LIMIT = 244,
    PROP_ACCESS_ALARM_EVENTS = 245,
    PROP_ACCESS_DOORS = 246,
    PROP_ACCESS_EVENT = 247,
    PROP_ACCESS_EVENT_AUTHENTICATION_FACTOR = 248,
    PROP_ACCESS_EVENT_CREDENTIAL = 249,
    PROP_ACCESS_EVENT_TIME = 250,
    PROP_ACCESS_RULES = 251,
    PROP_ACCESS_RULES_ENABLE = 252,
    PROP_ACCESS_TRANSACTION_EVENTS = 253,
    PROP_ACCOMPANIED = 254,
    PROP_ACTIVATION_TIME = 255,
    PROP_ACTIVE_AUTHENTICATION_POLICY = 256,
    PROP_ASSIGNED_ACCESS_RIGHTS = 257,
    PROP_AUTHENTICATION_FACTOR_INPUT_LIST = 258,
    PROP_AUTHENTICATION_FACTORS = 259,
    PROP_AUTHENTICATION_POLICY_LIST = 260,
    PROP_AUTHENTICATION_POLICY_NAMES = 261,
    PROP_AUTHORIZATION_MODE = 262,
    PROP_BELONGS_TO = 263,
    PROP_CREDENTIAL_DISABLE = 264,
    PROP_CREDENTIAL_STATUS = 265,
    PROP_CREDENTIALS = 266,
    PROP_CREDENTIALS_IN_ZONE = 267,
    PROP_DAYS_REMAINING = 268,
    PROP_ENTRY_POINTS = 269,
    PROP_EXIT_POINTS = 270,
    PROP_EXPIRY_TIME = 271,
    PROP_EXTENDED_TIME_ENABLE = 272,
    PROP_FAILED_ATTEMPT_EVENTS = 273,
    PROP_FAILED_ATTEMPTS = 274,
    PROP_FAILED_ATTEMPTS_TIME = 275,
    PROP_FORMAT_CLASS_SUPPORTED = 276,
    PROP_FORMAT_TYPE = 277,
    PROP_LAST_ACCESS_EVENT = 278,
    PROP_LAST_ACCESS_POINT = 279,
    PROP_LAST_CREDENTIAL_ADDED = 280,
    PROP_LAST_CREDENTIAL_ADDED_TIME = 281,
    PROP_LAST_CREDENTIAL_REMOVED = 282,
    PROP_LAST_CREDENTIAL_REMOVED_TIME = 283,
    PROP_LAST_USE_TIME = 284,
    PROP_LOCKDOWN = 285,
    PROP_LOCKDOWN_RELINQUISH_TIME = 286,
    PROP_MASTER_EXEMPTION = 287,
    PROP_MAX_FAILED_ATTEMPTS = 288,
    PROP_MEMBERS = 289,
    PROP_MASTER_POINT = 290,
    PROP_NUMBER_OF_AUTHENTICATION_POLICIES = 291,
    PROP_OCCUPANCY_COUNT = 293,
    PROP_OCCUPANCY_COUNT_ENABLE = 294,
    PROP_OCCUPANCY_COUNT_EXEMPTION = 295,
    PROP_OCCUPANCY_LOWER_THRESHOLD = 296,
    PROP_OCCUPANCY_LOWER_THRESHOLD_ENFORCED = 297,
    PROP_OCCUPANCY_STATE = 298,
    PROP_OCCUPANCY_UPPER_LIMIT = 299,
    PROP_OCCUPANCY_UPPER_LIMIT_ENFORCED = 300,
    PROP_PASSBACK_EXEMPTION = 301,
    PROP_PASSBACK_MODE = 302,
    PROP_PASSBACK_TIMEOUT = 303,
    PROP_POSITIVE_ACCESS_RULES = 304,
    PROP_READ_STATUS = 305,
    PROP_REASON_FOR_DISABLE = 306,
    PROP_THREAT_AUTHORITY = 307,
    PROP_THREAT_LEVEL = 308,
    PROP_TRACE_FLAG = 309,
    PROP_TRANSACTION_NOTIFICATION_CLASS = 310,
    PROP_USER_EXTERNAL_IDENTIFIER = 311,
    /* enumerations 312-313 are used in Addendum k to ANSI/ASHRAE 135-2004 */
    PROP_CHARACTER_SET = 312,
    PROP_STRICT_CHARACTER_MODE = 313,
    /* enumerations 312-313 are used in Addendum k to ANSI/ASHRAE 135-2004 */
    PROP_BACKUP_AND_RESTORE_STATE = 314,
    PROP_BACKUP_PREPARATION_TIME = 315,
    PROP_RESTORE_PREPARATION_TIME = 316,
    /* enumerations 317-323 are used in Addendum j to ANSI/ASHRAE 135-2004 */
    PROP_USER_INFORMATION_REFERENCE = 317,
    PROP_USER_NAME = 318,
    PROP_USER_TYPE = 319,
    PROP_USES_REMAINING = 320,
    PROP_VENDOR_FORMAT_IDENTIFIER = 321,
    PROP_ZONE_FROM = 322,
    PROP_ZONE_TO = 323,
    /* enumerations 324-325 are used in Addendum i to ANSI/ASHRAE 135-2004 */
    PROP_BINARY_ACTIVE_VALUE = 324,
    PROP_BINARY_INACTIVE_VALUE = 325
        /* The special property identifiers all, optional, and required  */
        /* are reserved for use in the ReadPropertyConditional and */
        /* ReadPropertyMultiple services or services not defined in this standard.
*/
        /* Enumerated values 0-511 are reserved for definition by ASHRAE.  */
        /* Enumerated values 512-4194303 may be used by others subject to the  */
        /* procedures and constraints described in Clause 23.  */
} BACNET_PROPERTY_ID;
```

E.2 BACnet Engineering Units Codes

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

```
typedef enum {
    /* Acceleration */
    UNITS_METERS_PER_SECOND_PER_SECOND = 166,
    /* Area */
    UNITS_SQUARE_METERS = 0,
    UNITS_SQUARE_CENTIMETERS = 116,
    UNITS_SQUARE_FEET = 1,
    UNITS_SQUARE_INCHES = 115,
    /* Currency */
    UNITS_CURRENCY1 = 105,
    UNITS_CURRENCY2 = 106,
    UNITS_CURRENCY3 = 107,
    UNITS_CURRENCY4 = 108,
    UNITS_CURRENCY5 = 109,
    UNITS_CURRENCY6 = 110,
    UNITS_CURRENCY7 = 111,
    UNITS_CURRENCY8 = 112,
    UNITS_CURRENCY9 = 113,
    UNITS_CURRENCY10 = 114,
    /* Electrical */
    UNITS_MILLIAMPERES = 2,
    UNITS_AMPERES = 3,
    UNITS_AMPERES_PER_METER = 167,
    UNITS_AMPERES_PER_SQUARE_METER = 168,
    UNITS_AMPERE_SQUARE_METERS = 169,
    UNITS_FARADS = 170,
    UNITS_HENRYS = 171,
    UNITS_OHMS = 4,
    UNITS_OHM_METERS = 172,
    UNITS_MILLIOHMS = 145,
    UNITS_KILOHMS = 122,
    UNITS_MEGOHMS = 123,
    UNITS_SIEMENS = 173, /* 1 mho equals 1 siemens */
    UNITS_SIEMENS_PER_METER = 174,
    UNITS_TESLAS = 175,
    UNITS_VOLTS = 5,
    UNITS_MILLIVOLTS = 124,
    UNITS_KILOVOLTS = 6,
    UNITS_MEGAVOLTS = 7,
    UNITS_VOLT_AMPERES = 8,
    UNITS_KILOVOLT_AMPERES = 9,
    UNITS_MEGAVOLT_AMPERES = 10,
    UNITS_VOLT_AMPERES_REACTIVE = 11,
    UNITS_KILOVOLT_AMPERES_REACTIVE = 12,
    UNITS_MEGAVOLT_AMPERES_REACTIVE = 13,
    UNITS_VOLTS_PER_DEGREE_KELVIN = 176,
    UNITS_VOLTS_PER_METER = 177,
    UNITS_DEGREES_PHASE = 14,
    UNITS_POWER_FACTOR = 15,
    UNITS_WEBERS = 178,
    /* Energy */
    UNITS_JOULES = 16,
    UNITS_KILOJOULES = 17,
    UNITS_KILOJOULES_PER_KILOGRAM = 125,
    UNITS_MEGAJOULES = 126,
    UNITS_WATT_HOURS = 18,
    UNITS_KILOWATT_HOURS = 19,
    UNITS_MEGAWATT_HOURS = 146,
    UNITS_BTUS = 20,
    UNITS_KILO_BTUS = 147,
    UNITS_MEGA_BTUS = 148,
    UNITS_THERMS = 21,
    UNITS_TON_HOURS = 22,
    /* Enthalpy */
    UNITS_JOULES_PER_KILOGRAM_DRY_AIR = 23,
    UNITS_KILOJOULES_PER_KILOGRAM_DRY_AIR = 149,
    UNITS_MEGAJOULES_PER_KILOGRAM_DRY_AIR = 150,
    UNITS_BTUS_PER_POUND_DRY_AIR = 24,
    UNITS_BTUS_PER_POUND = 117,
    /* Entropy */
    UNITS_JOULES_PER_DEGREE_KELVIN = 127,
    UNITS_KILOJOULES_PER_DEGREE_KELVIN = 151,
    UNITS_MEGAJOULES_PER_DEGREE_KELVIN = 152,
    UNITS_JOULES_PER_KILOGRAM_DEGREE_KELVIN = 128,
    /* Force */
    UNITS_NEWTON = 153,
    /* Frequency */
    UNITS_CYCLES_PER_HOUR = 25,
    UNITS_CYCLES_PER_MINUTE = 26,
    UNITS_HERTZ = 27,
    UNITS_KILOHERTZ = 129,
    UNITS_MEGAHERTZ = 130,
    UNITS_PER_HOUR = 131,
    /* Humidity */
    UNITS_GRAMS_OF_WATER_PER_KILOGRAM_DRY_AIR = 28,
    UNITS_PERCENT_RELATIVE_HUMIDITY = 29,
    /* Length */
    UNITS_MILLIMETERS = 30,
    UNITS_CENTIMETERS = 118,
    UNITS_METERS = 31,
    UNITS_INCHES = 32,
    UNITS_FEET = 33,
    /* Light */
    UNITS_CANDELAS = 179,
    UNITS_CANDELAS_PER_SQUARE_METER = 180,
    UNITS_WATTS_PER_SQUARE_FOOT = 34,
    UNITS_WATTS_PER_SQUARE_METER = 35,
    UNITS_LUMENS = 36,
    UNITS_LUXES = 37,
    UNITS_FOOT_CANDLES = 38,
    /* Mass */
    UNITS_KILOGRAMS = 39,
    UNITS_POUNDS_MASS = 40,
    UNITS_TONS = 41,
    /* Mass Flow */
    UNITS_GRAMS_PER_SECOND = 154,
    UNITS_GRAMS_PER_MINUTE = 155,
    UNITS_KILOGRAMS_PER_SECOND = 42,
    UNITS_KILOGRAMS_PER_MINUTE = 43,
    UNITS_KILOGRAMS_PER_HOUR = 44,
    UNITS_POUNDS_MASS_PER_SECOND = 119,
    UNITS_POUNDS_MASS_PER_MINUTE = 45,
    UNITS_POUNDS_MASS_PER_HOUR = 46,
    UNITS_TONS_PER_HOUR = 156,
    /* Power */
    UNITS_MILLIWATTS = 132,
    UNITS_WATTS = 47,
    UNITS_KILOWATTS = 48,
    UNITS_MEGAWATTS = 49,
    UNITS_BTUS_PER_HOUR = 50,
    UNITS_KILO_BTUS_PER_HOUR = 157,
    UNITS_HORSEPOWER = 51,
    UNITS_TONS_REFRIGERATION = 52,
    /* Pressure */
    UNITS_PASCALS = 53,
    UNITS_HECTOPASCALS = 133,
    UNITS_KILOPASCALS = 54,
    UNITS_MILLIBARS = 134,
    UNITS_BARS = 55,
    UNITS_POUNDS_FORCE_PER_SQUARE_INCH = 56,
    UNITS_CENTIMETERS_OF_WATER = 57,
    UNITS_INCHES_OF_WATER = 58,
    UNITS_MILLIMETERS_OF_MERCURY = 59,
    UNITS_CENTIMETERS_OF_MERCURY = 60,
    UNITS_INCHES_OF_MERCURY = 61,
    /* Temperature */
    UNITS_DEGREES_CELSIUS = 62,
    UNITS_DEGREES_KELVIN = 63,
    UNITS_DEGREES_KELVIN_PER_HOUR = 181,
    UNITS_DEGREES_KELVIN_PER_MINUTE = 182,
    UNITS_DEGREES_FAHRENHEIT = 64,
    UNITS_DEGREE_DAYS_CELSIUS = 65,
    UNITS_DEGREE_DAYS_FAHRENHEIT = 66,
    UNITS_DELTA_DEGREES_FAHRENHEIT = 120,
    UNITS_DELTA_DEGREES_KELVIN = 121,
    /* Time */
    UNITS_YEARS = 67,
    UNITS_MONTHS = 68,
    UNITS_WEEKS = 69,
    UNITS_DAYS = 70,
    UNITS_HOURS = 71,
    UNITS_MINUTES = 72,
    UNITS_SECONDS = 73,
    UNITS_HUNDREDTHS_SECONDS = 158,
    UNITS_MILLISECONDS = 159,
    /* Torque */
    UNITS_NEWTON_METERS = 160,
    /* Velocity */
    UNITS_MILLIMETERS_PER_SECOND = 161,
    UNITS_MILLIMETERS_PER_MINUTE = 162,
    UNITS_METERS_PER_SECOND = 74,
    UNITS_METERS_PER_MINUTE = 163,
    UNITS_METERS_PER_HOUR = 164,
    UNITS_KILOMETERS_PER_HOUR = 75,
    UNITS_FEET_PER_SECOND = 76,
    UNITS_FEET_PER_MINUTE = 77,
    UNITS_MILES_PER_HOUR = 78,
    /* Volume */
    UNITS_CUBIC_FEET = 79,
    UNITS_CUBIC_METERS = 80,
    UNITS_IMPERIAL_GALLONS = 81,
    UNITS_LITERS = 82,
    UNITS_US_GALLONS = 83,
    /* Volumetric Flow */
    UNITS_CUBIC_FEET_PER_SECOND = 142,
    UNITS_CUBIC_FEET_PER_MINUTE = 84,
    UNITS_CUBIC_METERS_PER_SECOND = 85,
    UNITS_CUBIC_METERS_PER_MINUTE = 165,
    UNITS_CUBIC_METERS_PER_HOUR = 135,
    UNITS_IMPERIAL_GALLONS_PER_MINUTE = 86,
    UNITS_LITERS_PER_SECOND = 87,
    UNITS_LITERS_PER_MINUTE = 88,
    UNITS_LITERS_PER_HOUR = 136,
    UNITS_US_GALLONS_PER_MINUTE = 89,
    /* Other */
    UNITS_DEGREES_ANGULAR = 90,
    UNITS_DEGREES_CELSIUS_PER_HOUR = 91,
    UNITS_DEGREES_CELSIUS_PER_MINUTE = 92,
    UNITS_DEGREES_FAHRENHEIT_PER_HOUR = 93,
    UNITS_DEGREES_FAHRENHEIT_PER_MINUTE = 94,
    UNITS_JOULE_SECONDS = 183,
    UNITS_KILOGRAMS_PER_CUBIC_METER = 186,
    UNITS_KW_HOURS_PER_SQUARE_METER = 137,
    UNITS_KW_HOURS_PER_SQUARE_FOOT = 138,
    UNITS_MEGAJOULES_PER_SQUARE_METER = 139,
    UNITS_MEGAJOULES_PER_SQUARE_FOOT = 140,
    UNITS_NO_UNITS = 95,
    UNITS_NEWTON_SECONDS = 187,
    UNITS_NEWTONS_PER_METER = 188,
    UNITS_PARTS_PER_MILLION = 96,
    UNITS_PARTS_PER_BILLION = 97,
    UNITS_PERCENT = 98,
    UNITS_PERCENT_OBSCURATION_PER_FOOT = 143,
    UNITS_PERCENT_OBSCURATION_PER_METER = 144,
    UNITS_PERCENT_PER_SECOND = 99,
    UNITS_PER_MINUTE = 100,
    UNITS_PER_SECOND = 101,
    UNITS_PSI_PER_DEGREE_FAHRENHEIT = 102,
    UNITS_RADIANS = 103,
    UNITS_RADIANS_PER_SECOND = 184,
    UNITS_REVOLUTIONS_PER_MINUTE = 104,
    UNITS_SQUARE_METERS_PER_NEWTON = 185,
    UNITS_WATTS_PER_METER_PER_DEGREE_KELVIN = 189,
    UNITS_WATTS_PER_SQUARE_METER_DEGREE_KELVIN = 141,
       ; /* Enumerated values 0-255 are reserved for definition by ASHRAE. */
        /* Enumerated values 256-65535 may be used by others subject to */
        /* the procedures and constraints described in Clause 23. */
        /* The last enumeration used in this version is 189. */
    MAX_UNITS = 190
} BACNET_ENGINEERING_UNITS;
```

Appendix F LonWorks XIF File Format

F.1 XIF Files

The XIF file may be exported via the configuration tool for those network management tools that explicitly require a file. The content of the XIF file may also be extracted from the device itself by tools such as LonMaker. The only difference between file export and extracting the XIF from the device is that the user given network variable names are not stored in the device. If the XIF is extracted from the device by a network management tool, generic names as originally compiled into code are provided.

The XIF file follows the version 4.402 specification available from LonMark at www.lonmark.org. You will need to obtain a copy of that specification from LonMark if you have a need to study the XIF file in detail.

The XIF file is a text file with a .xif suffix and the file may be opened with any text editor. The beginning of the file will appear as follows, illustrated here in part so that you can identify what an XIF file looks like. **Do not attempt to edit this file!**

```
File: objlist2.xif generated by BB2-3020 Tool v2.03, XIF Version 4.402
Copyright (c) Control Solutions, Inc. 2014
All Rights Reserved. Run on Mon Mar 23 12:21:22 2015

80:00:17:47:1E:84:04:01
2 15 1 243 0 8 8 5 5 8 8 11 11 11 11 7 243 16 127 0 1 128 243 0 0 0 0 0 0 0 0 0 2 15
0 0 0 0 0 2 1 0 0
32 7 19 13 28 0 0 15 5 3 841 4 10000000 1
1 7 1 1 4 4 4 15 200 0
78125 0 0 0 0 0 252 0 0 0 0 0
90 0 240 0 0 0 40 40 0 5 22 9 26 43 44
*
"&3.4@0nodeObject,1[120openLoopSensor,3[120openLoopActuator

VAR nviRequest 0 0 0 0
0 1 63 0 0 1 0 1 0 1 0 0 0
"@0|1
92 * 2
2 0 0 0 0
1 0 0 1 0
VAR nvoStatus 1 0 0 0
0 1 63 1 0 1 0 1 0 1 0 1 0
"@0|2
93 * 26
2 0 0 0 0
3 0 1 0 0
3 1 1 0 0
3 2 1 0 0
3 3 1 0 0
3 4 1 0 0
3 5 1 0 0
3 6 1 0 0
3 7 1 0 0
3 0 1 0 0
3 1 1 0 0
3 2 1 0 0
3 3 1 0 0
3 4 1 0 0
```

```
3 5 1 0 0
3 6 1 0 0
3 7 1 0 0
3 0 1 0 0
3 1 1 0 0
3 2 1 0 0
3 3 1 0 0
3 4 1 0 0
3 5 1 0 0
3 6 1 0 0
3 7 1 0 0
3 0 8 0 0
VAR nvoFileDirectory 2 0 0 0
0 1 63 1 0 1 0 1 0 1 1 0 0
"@0|8
114 * 1
2 0 0 0 0
VAR Test_Object_1 3 0 0 0
0 1 63 1 0 1 0 1 0 1 0 0 0
"@1|1?
51 * 1
4 0 4 0 0
VAR Test_Object_2 4 0 0 0
0 1 63 1 0 1 0 1 0 1 0 0 0
"@2|1?
51 * 1
4 0 4 0 0
VAR nvoSenValue_3 5 0 0 0
0 1 63 1 0 1 0 1 0 1 0 0 0
"@3|1?
0 * 1
1 0 0 0 31
VAR nvoSenValue_4 6 0 0 0
0 1 63 1 0 1 0 1 0 1 0 0 0
"@4|1?
0 * 1
1 0 0 0 31
```

Appendix G Configuration XML File Format

G.1 Configuration Files

The configuration file that is used to save gateway configuration and reload later to reconfigure another gateway the same way is saved in XML format. This makes it easy to read for diagnostic purposes, primarily for Control Solutions' technical support use. However, due to the complexity and interaction between the parts of the file,
the **XML FILE IS NOT INTENDED TO BE MANUALLY EDITED**.

There is no reason to manually edit an XML file. If you are looking for a short cut in configuring the gateway, you are looking in the wrong direction. The configuration tool includes the ability to import and export object lists as a CSV file, and import and export XIF files. Manual editing should be limited to creating and modifying the CSV file. Once your files are imported, you can auto-create much of the configuration.

The configuration software (user interface) includes a number of error checking steps, and these are bypassed in the event you manually edit an XML file. **IF YOU HAVE CREATED PROBLEMS BY MANUALLY EDITING AN XML FILE, CONTROL SOLUTIONS WILL NOT HELP YOU FIX IT.**

Appendix H USB Driver Installation

H.1 Driver Installation

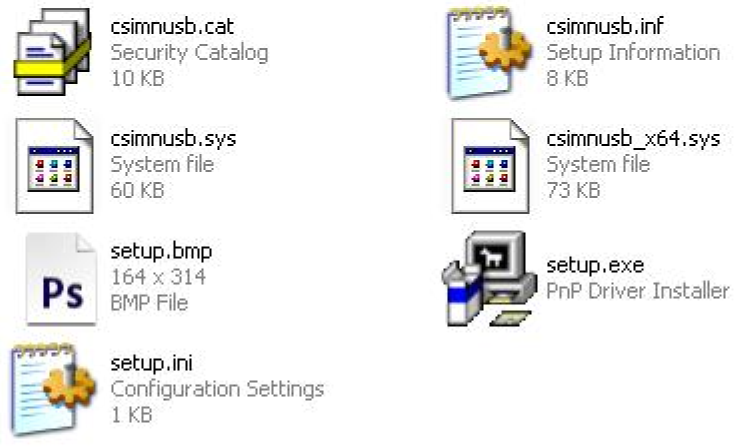
The required USB driver used to be included as a standard part of Windows, and all that needed to be done to “install” the driver was provide a configuration file telling Windows which driver to use. Starting with Windows 7, that driver was no longer included, and Windows 8 complicated matters even further.

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

The USB driver provided in Control Solutions’ driver package will install in Windows XP, 7, and 8, and both 32-bit and 64-bit versions. It includes the necessary driver signing verified through Verisign (now part of Symantec).

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

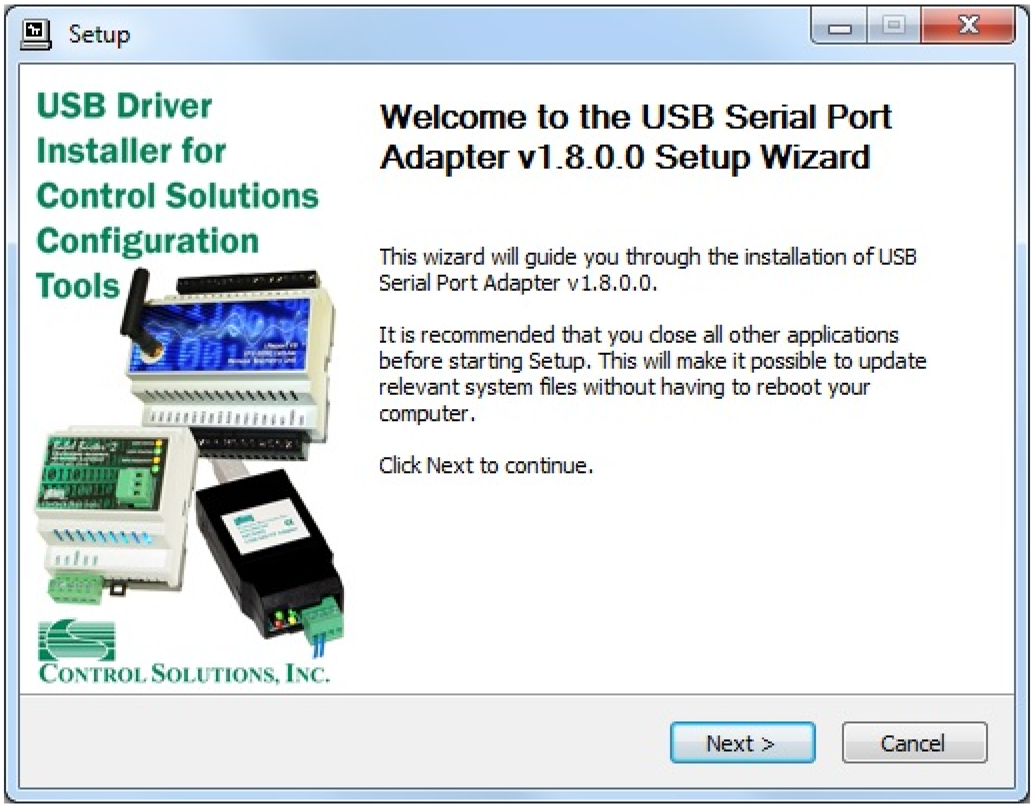
Unzip the contents of this file into a directory somewhere on your PC. The contents will look like this:



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

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

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

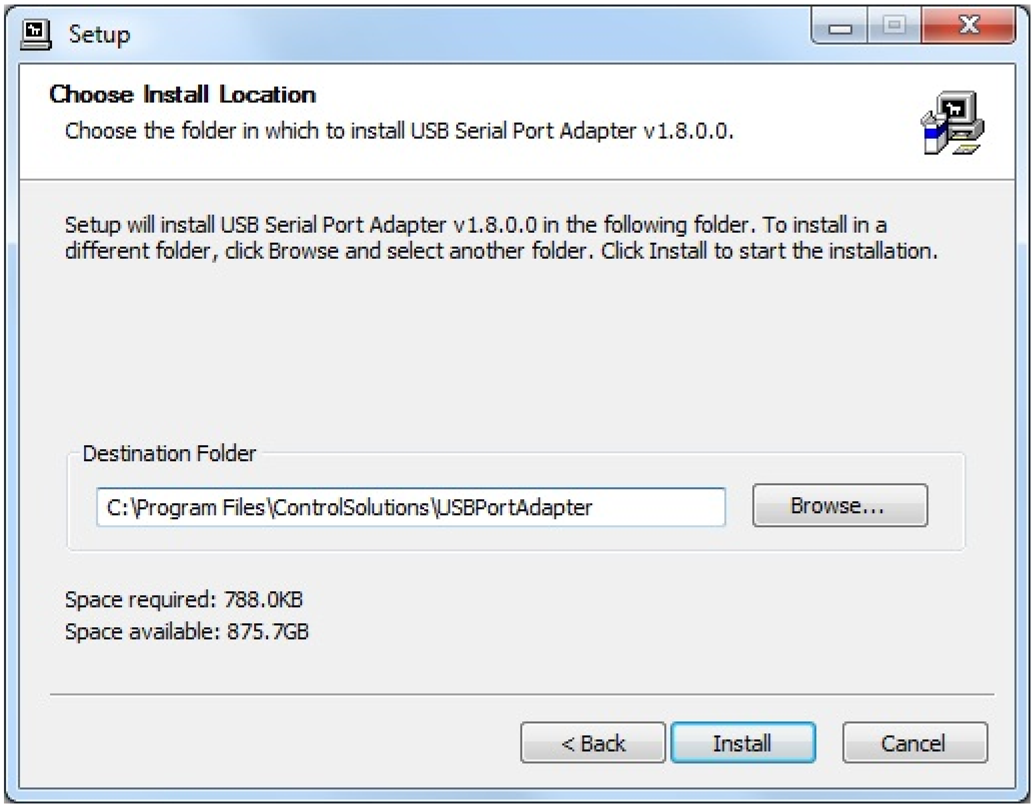

Setup
USB Driver Installer for Control Solutions Configuration Tools
CONTROL SOLUTIONS, INC.
Next >
Cancel

Welcome to the USB Serial Port Adapter v1.8.0.0 Setup Wizard

This wizard will guide you through the installation of USB Serial Port Adapter v1.8.0.0.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.



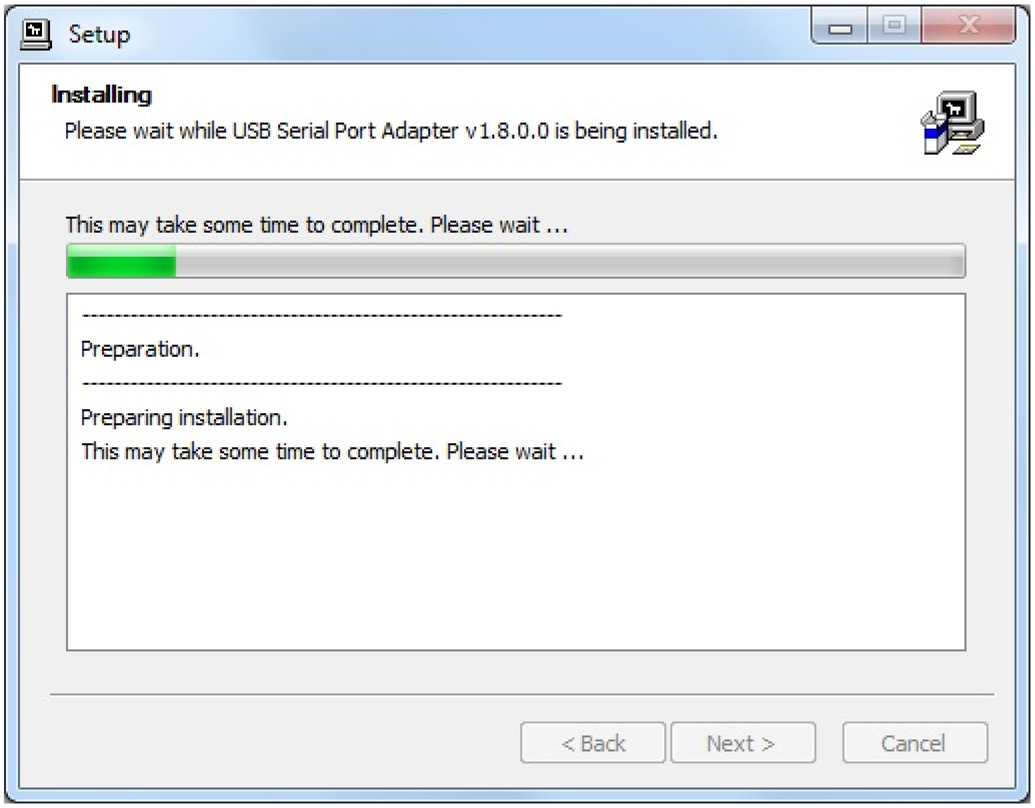
Choose Install Location

Choose the folder in which to install USB Serial Port Adapter v1.8.0.0.

Setup will install USB Serial Port Adapter v1.8.0.0 in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files\ControlSolutions\USBPortAdapter

Space required: 788.0KB
Space available: 875.7GB



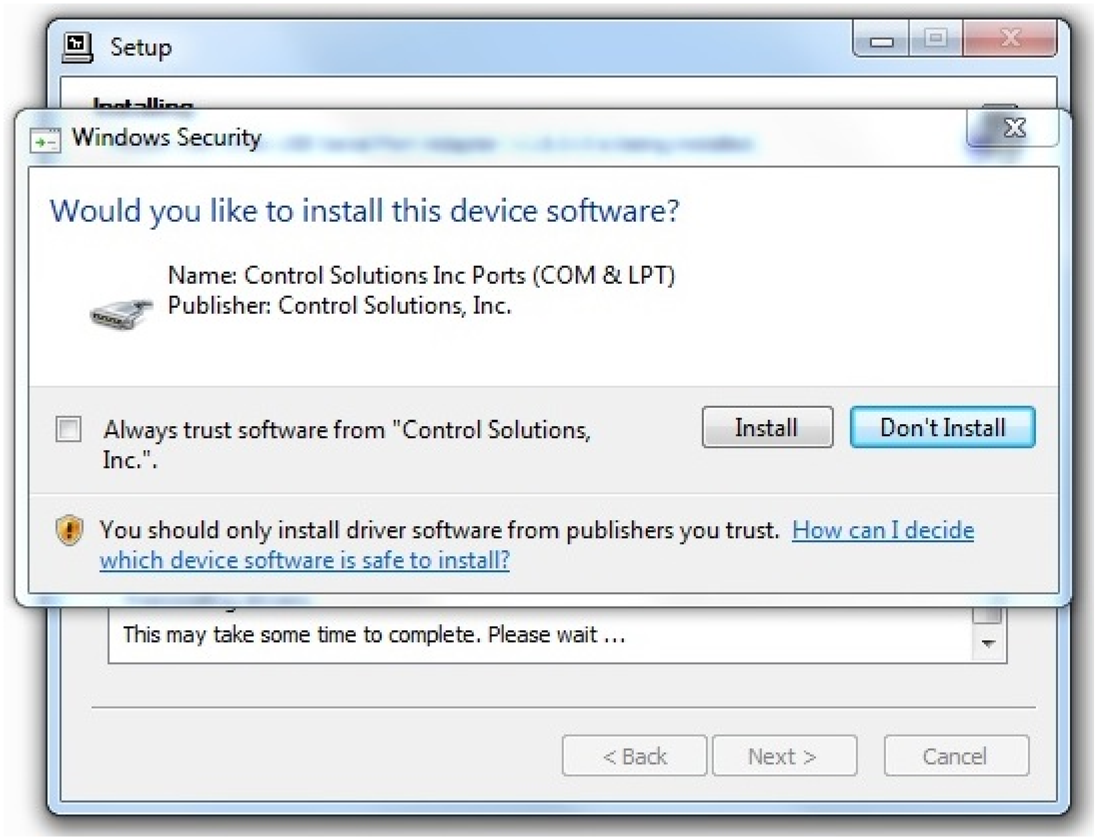
Installing

Please wait while USB Serial Port Adapter v1.8.0.0 is being installed.

This may take some time to complete. Please wait ...

```
-------------------------------------------------------------
Preparation.
-------------------------------------------------------------
Preparing installation.
This may take some time to complete. Please wait ...
```

Click on “Install” when you get to this window:



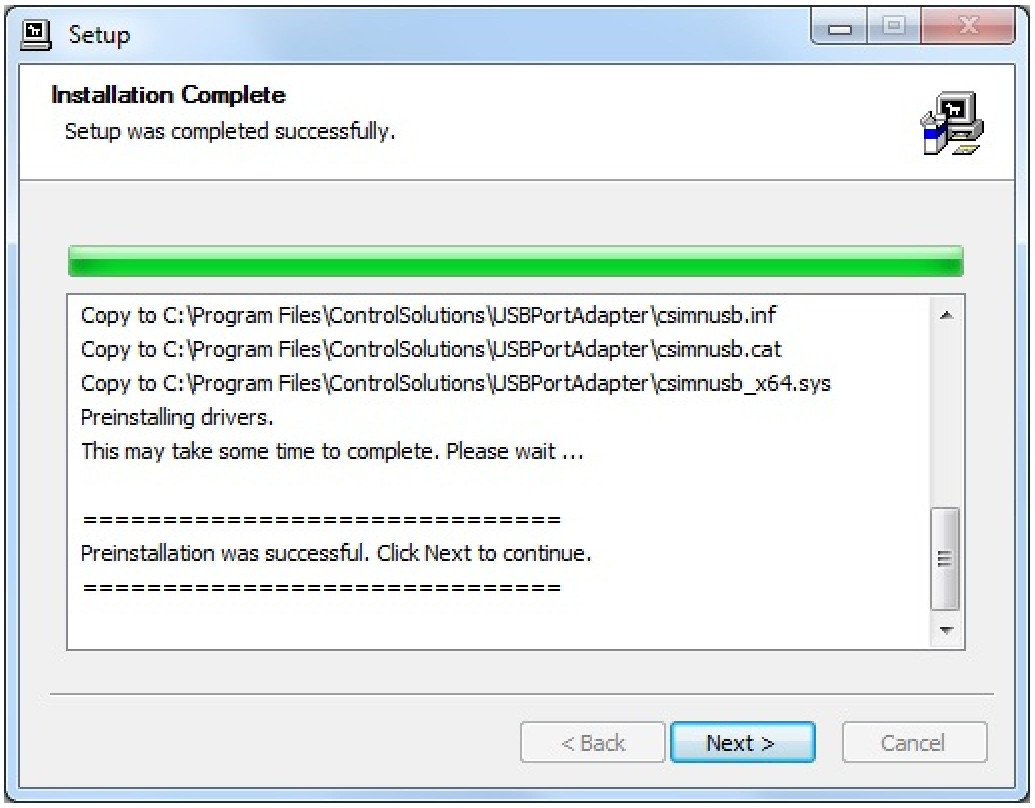
Would you like to install this device software?

Name: Control Solutions Inc Ports (COM & LPT)
Publisher: Control Solutions, Inc.

Always trust software from "Control Solutions, Inc.".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

This may take some time to complete. Please wait ...



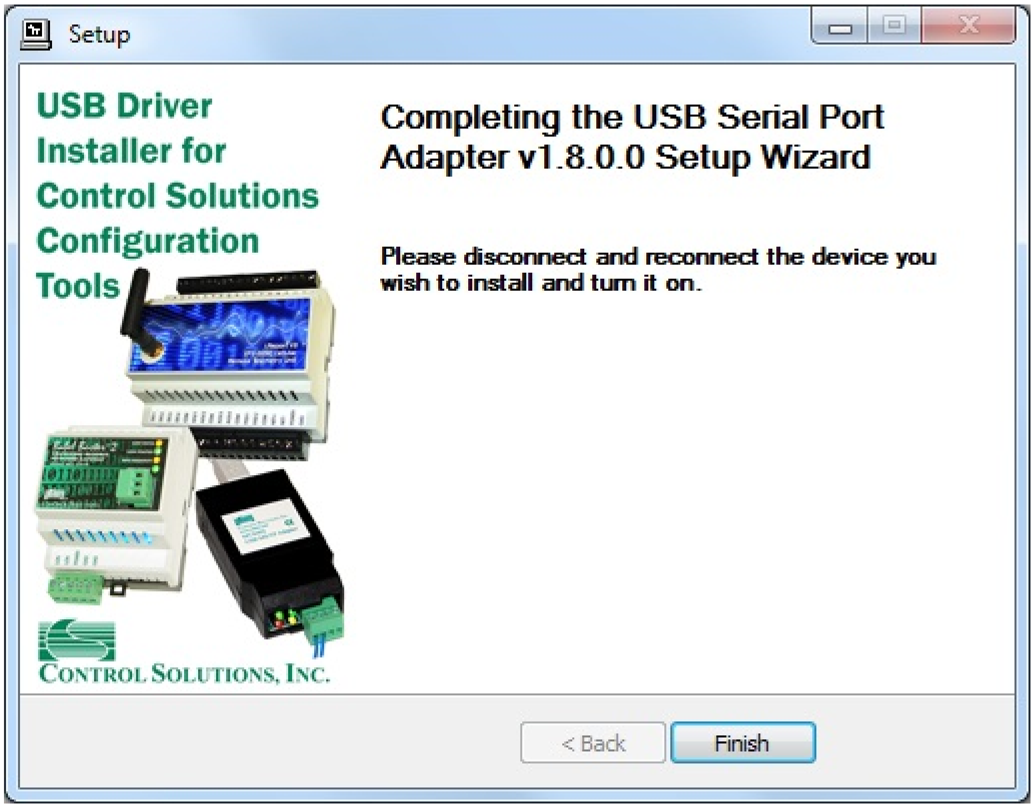
Installation Complete
Setup was completed successfully.

```
Copy to C:\Program Files\ControlSolutions\USBPortAdapter\csimnusb.inf
Copy to C:\Program Files\ControlSolutions\USBPortAdapter\csimnusb.cat
Copy to C:\Program Files\ControlSolutions\USBPortAdapter\csimnusb_x64.sys
Preinstalling drivers.
This may take some time to complete. Please wait ...

================================
Preinstallation was successful. Click Next to continue.
================================
```

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

Solutions configuration tool's "Connect" page.


Setup
USB Driver Installer for Control Solutions Configuration Tools
Control Solutions, Inc.
< Back
Finish

Completing the USB Serial Port Adapter v1.8.0.0 Setup Wizard

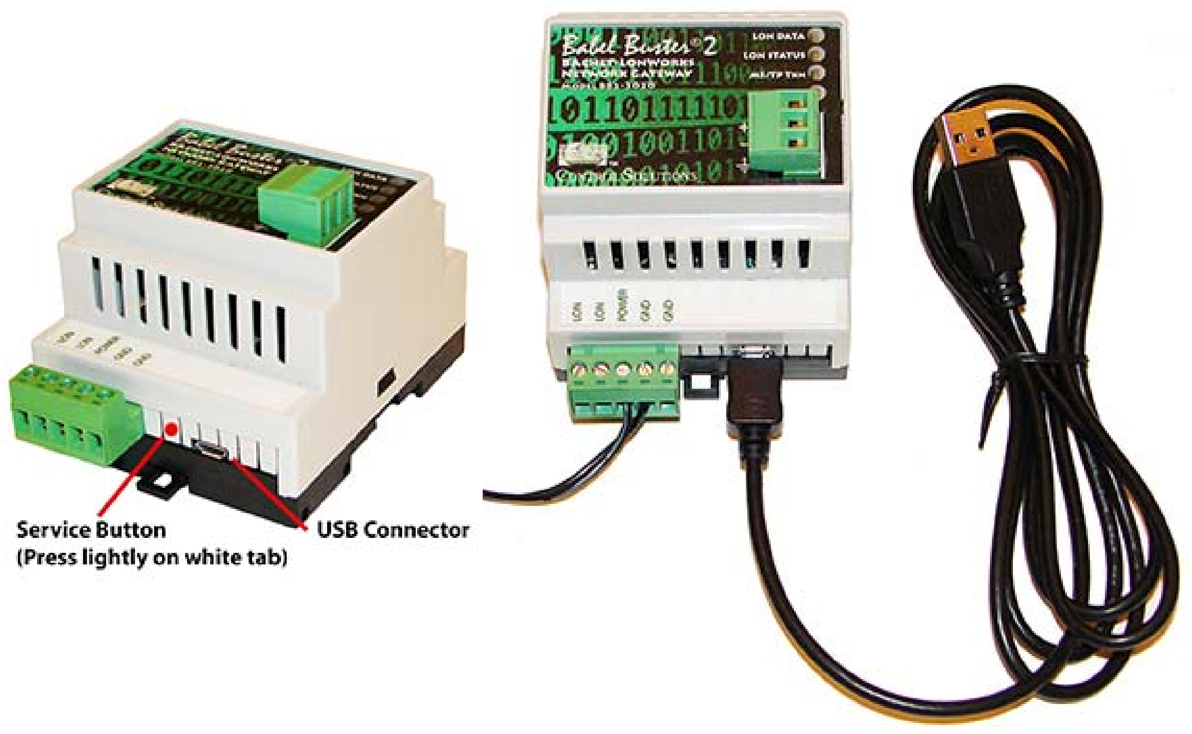
Please disconnect and reconnect the device you wish to install and turn it on.

Appendix J Hardware Details

J.1 Service Button & USB Connection

Connect USB cable as illustrated. Install the driver package provided by Control Solutions prior to attempting to use the USB connection.

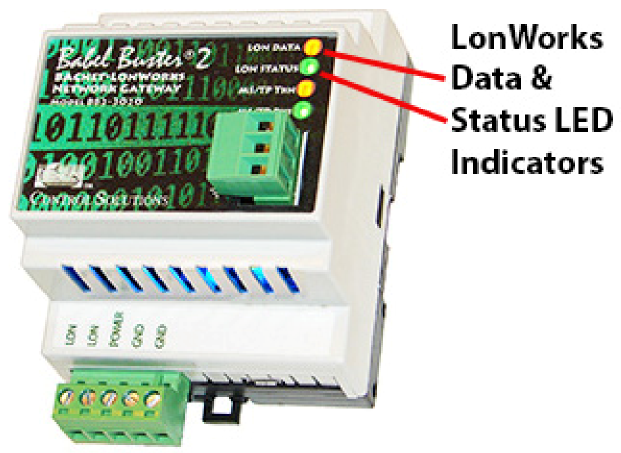
The service button is hidden behind the tab illustrated below. Press lightly on the white tab indicated. You will feel a slight clicking action when the button is pressed. It is not necessary to remove the cover to press the button - it is intended to be actuated by the plastic tab that is part of the cover.

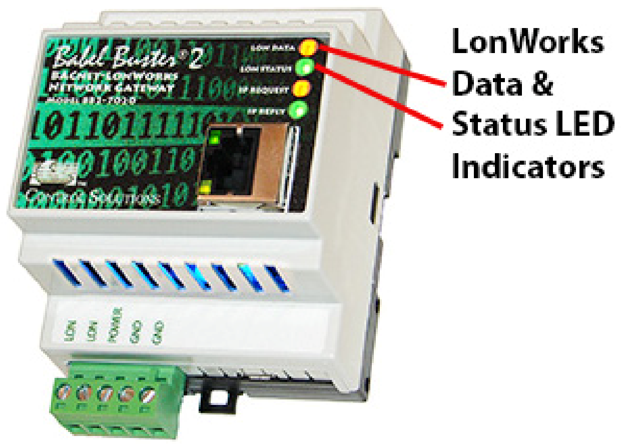


J.2 Front Panel LED Indicators

Power-up LED behavior for BB2-3020 MS/TP gateway: All LEDs on front panel will turn on yellow or red for half a second, then all will turn on green for half a second. Then they will proceed to indicate as normally defined for the indicators.

Power-up LED behavior for BB2-7020 IP gateway: Will behave the same as MS/TP, except the IP request/reply behavior will be delayed by several seconds after power-up indicaton on the LonWorks LEDs.



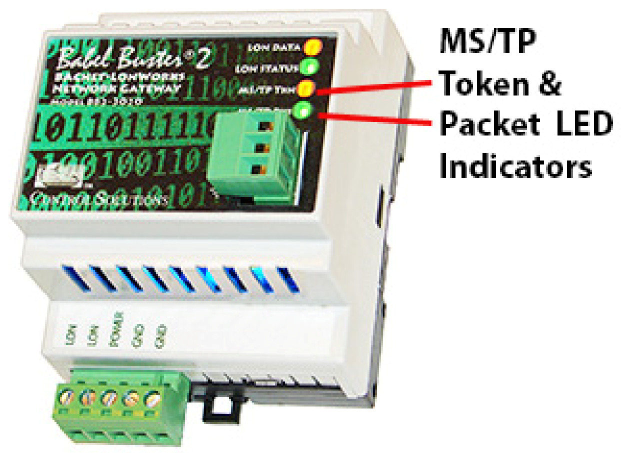


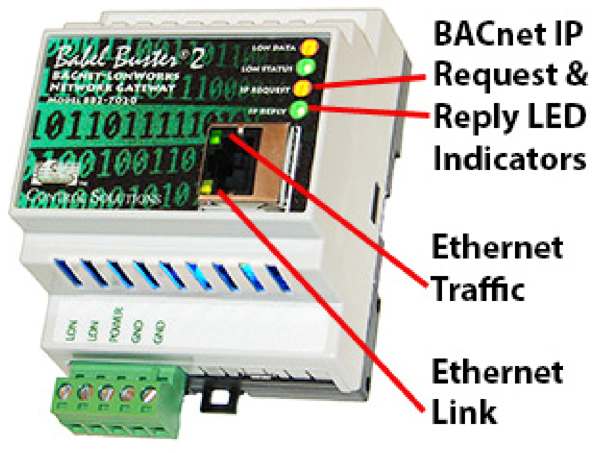
The LonWorks indicator LEDs display LonWorks network activity. Note, however, that activity of these LEDs is not only affected by configuration of the device, but by whether the device is commissioned on the network and whether other devices are also communicating.

Mode	Data LED	Status LED
Wink	Alternates between yellow & green 10 times, then resume normal mode.	Alternates between red & green 10 times, then resume normal mode.
Normal	Yellow flash indicates NV update was sent by gateway. Green flash indicates NV update was received by gateway.	Red indicates Neuron chip is not running. Green indicates gateway's host processor is communicating with LonWorks Neuron chip.

The "wink" behavior is invoked by sending a wink command to the BB2 gateway via the LonWorks network. This is generally just a diagnostic to see if you are successfully communicating with the device via LonWorks. Other than the few seconds it takes to execute the wink, the device will always be in "normal" mode as far as LED indications are concerned in the table above.

The LED indicators on the front of the gateway also indicate BACnet errors. These are global indicators that do not tell you which device or which object is having trouble, but these indicators are a very quick way to observe whether there are problems, and also whether there is any activity at all.





The MS/TP or IP LEDs will indicate BACnet traffic as indicated in the tables below.

BB2-3020 MS/TP LEDs		
Mode	MS/TP TKN (token) LED	MS/TP PKT (packet) LED
Gateway is Client or Server	Flash yellow each time the gateway sends a "poll for master". Flash green each time the token is passed.	Flash green any time a packet is sent on the network. This can be either a request or a response to a request Flash red any time an error is detected. The error can be an error code reply, or timeout. The object reliability code will indicate which one it was.

An MS/TP device that is functioning normally will always be at least passing the token, and usually polling for master periodically. The only time a device will not poll for master is if another device exists on the network with a MAC address only 1 greater than the gateway itself. It then just passes the token to that device without any intermediate polling for master.

Two particular indications of the MS/TP token LED are worth noting.

If the token LED appears to be on nearly solid yellow (or amber), this means the gateway is doing nothing but poll for token, which implies it is not finding any other devices on the network. You will see this any time you apply power to the gateway without connecting any network. If the network is connected, there is a problem with connections or port settings.

If the token LED is completely off and never flashes on, this means the gateway is hearing other traffic on the network, but just not connecting. An MS/TP device is required to "be quiet until spoken to". Therefore, if there is chatter on the network, but the gateway cannot find anything that is addressed to the gateway, it will remain quiet. This problem is usually the result of problems with port settings.

A gateway (or any MS/TP device) can alternate between the above two conditions. The "be quiet until spoken to" rule is up for grabs when the entire network is powered up simultaneously. Whoever wakes up first will be generating all the chatter whiile everybody else on the network listens.

If the token LED is very intermittent, flashes rapidly a few times (including green on the token LED), then is off completely for a period of time, this usually means it is trying to talk, thought it was talking or actually was for a bit, but something got out of sync. If there are duplicate device addresses on the network or mismatching Max Master settings, this will result in the appearance of intermittent communications.

Check you wiring. Noise on the line resulting from not following RS-485 wiring guidelines will result in communication problems ranging from no connection at all to frequent errors. Also be sure taht you have a ground connection between all devices on the network. The RS-485 network is not a 2-wire network. You need two data lines plus signal ground. If you consider cable shield a connection, then you need 4 connections.

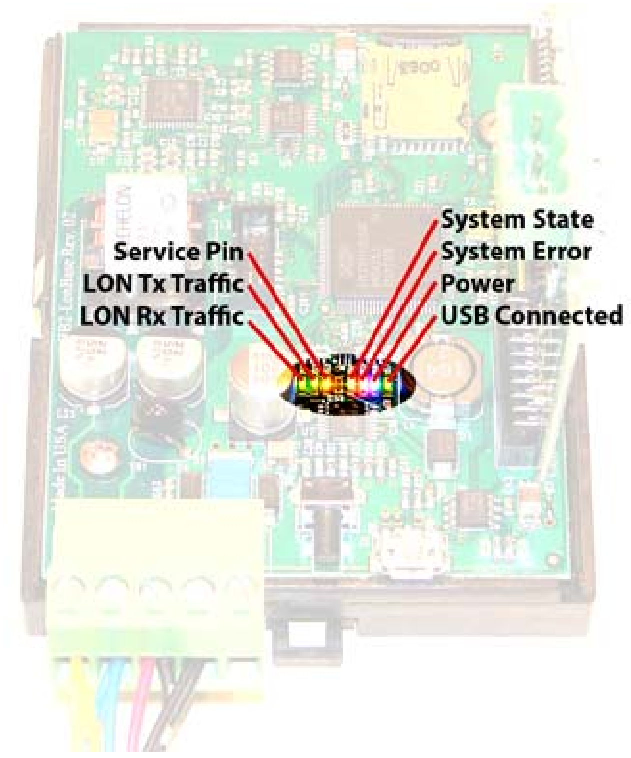
BB2-7020 IP LEDs		
Mode	IP Request LED	IP Reply LED
Gateway is Client (Master)	Flash yellow each time gateway sends a request to a remote server (slave).	Flash green when gateway receives a good response. Flash red when gateway receives error code message from server, or if timed out with no response from server.
Gateway is Server (Slave)	Flash yellow each time gateway receives a request from external client.	Flash green when gateway recognizes request as good/valid and sends a good reply. Flash red when gateway receives a request that results in replying with an error code.

The Ethernet traffic LED will indicate any traffic on the Ethernet network, and does not necessarily indicate BACnet IP traffic. The traffic LED will typically be off more than on, flashing on each time traffic is indicated. If the traffic LED is on completely solid, the server is not running (normal for a half minute or so during startup).

The Ethernet link LED will be on any time there is a connection to the network. If the Ethernet cable is unplugged, this light will go out. If BACnet IP is failing and this light is out, check Ethernet cables.

J.3 Internal Diagnostic LED Indicators

The internal diagnostic LEDs may be observed through the vent slots in the case. You normally have minimal need to observe these, but if you are having trouble, you may want to check these.



Service Pin	Flashes yellow any time service pin message is sent by LonWorks Neuron.
LON Tx Traffic	Flashes green when message is transmitted by LonWorks Neuron.
LON Rx Traffic	Flashes green when message is received by LonWorks Neuron.
System State	Flashes green codes indicating system state, normally on with brief flash off once every 2-3 seconds as heartbeat indicator.
System Error	Flashes red error codes if a hardware fault has been detected. Normally off.
Power	Blue, should always be on, indicates power is present.
USB Connected	Turns on green when USB connection is made with PC.

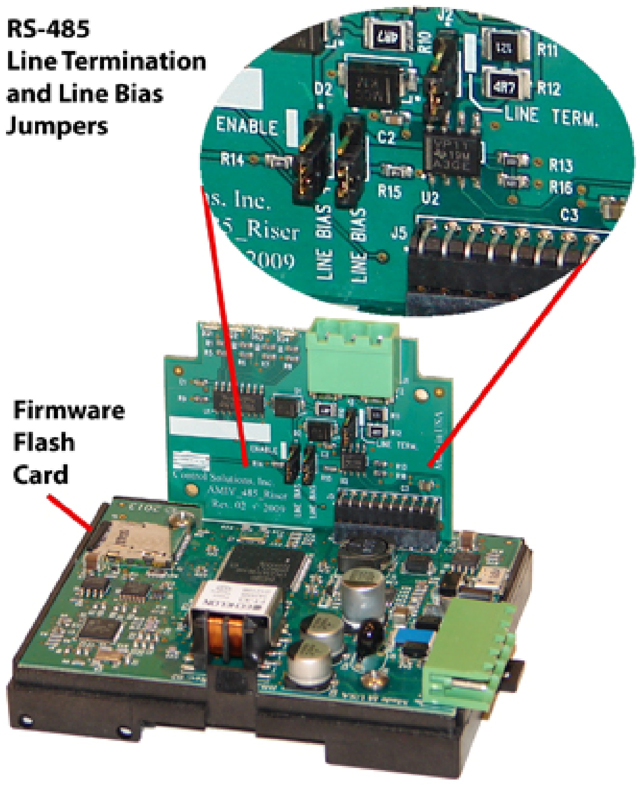
During power-up, the blue LED should turn on immediately, and most other LEDs will flash briefly. The system state and system error LEDs will deliberately flash once, and the system state LED will typically flash a short flash more than once before resuming normal heartbeat indication. If a hardware fault has been detected, the red system error LED will continue to flash a code. During a firmware update, the indicators take on a different set of meanings, and these will be provided along with update instructions as applicable. If any abnormal indications are observed, contact technical support (www.csimn.com/ticket) for additional advice.

J.4 RS-485 Line Termination & Bias

Enable line termination only when this device is placed at the end of the network. Termination should only be enabled at two points on the network, and these two points must be specifically the end points.

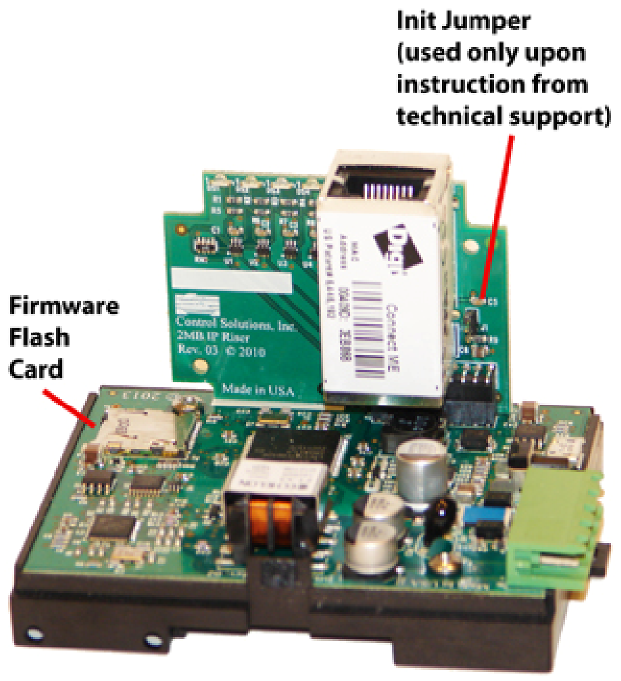
Enable line bias when needed. Line bias should only be enabled at one point on the network, and does not have to be the end point. Line bias holds the line in a known neutral state when no devices are transmitting. Without bias, the transition from offline to online by a transmitter can look like a false start bit and cause loss of communication.

The line conditioning options are enabled when the respective shunt is moved to the position indicated by the white block next to the 3-pin header. Putting the shunt on the opposite 2 pins disables the option, and is simply a place to store the shunt.



J.5 Server Module Init Jumper

The "Init" jumper on the server module should only be used when advised by tech support. Installing this jumper prior to power-up causes the server to go into firmware update mode.


Init Jumper (used only upon instruction from technical support)
Firmware Flash Card
Control Solutions, Inc.
2MB IP Riser
Rev. 03 © 2010
Made in USA
Connect ME