

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



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

User Guide

Babel Buster 2

© 2015 Control Solutions, Inc.

User Guide Contents

1 Introduction

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

2 Overview of Gateway Functions

- 2.1 Object Server Model for a Gateway
- 2.2 Data Flow in the Gateway
- 2.3 Non-Bound Polling of Network Variables

3 Overview of How to Configure Gateway

- 3.1 The Basics
- 3.2 Build Configuration from XIF File for LonWorks Device
- 3.3 Build Configuration by Importing XIF from LonWorks Device
- 3.4 Build Configuration from Scratch
- 3.5 Build Configuration from CSV List of BACnet Objects

4 Tool 'Connect' Page

- 4.1 Connecting Configuration Tool to Gateway Device

5 Tool 'Obj Import' Page

- 5.1 Importing a CSV Object List

6 Tool 'Obj List' Page

- 6.1 Auto-Building the Configuration
- 6.2 Editing the Object List
- 6.3 Object List Export
- 6.4 Definition of BACnet Object Configuration Parameters
- 6.5 Definition of BACnet Client Map Configuration Parameters

7 Tool 'NV Import' Page

- 7.1 Importing an XIF File

8 Tool 'NV List' Page

- 8.1 Configuration from XIF File
- 8.2 Building Configuration Manually
- 8.3 Using the NV Editor
- 8.4 Using NV CSV Files to Build Multi-Device Configuration

9 Tool 'Master List' Page

- 9.1 Editing Configuration from Master List Page
- 9.2 Sending Configuration To Device
- 9.3 Getting Configuration From Device
- 9.4 Fixing Conflicts

10 Tool 'View Data' Page

- 10.1 Viewing Object Data
- 10.2 Changing Object Data
- 10.3 Diagnostics - Object Reliability

11 Tool 'BACnet Port' Page

- 11.1 BACnet Device Settings
- 11.2 BACnet Object Counts
- 11.3 MS/TP Port Settings
- 11.4 IP Port Settings

12 Tool 'LonWorks' Page

- 12.1 Viewing LonWorks Identity of the Gateway
- 12.4 Changing Program ID of the Gateway
- 12.5 Viewing Identity and Status of Other LonWorks Devices
- 12.6 Node Discovery Using Service Pin

- 12.7 Discovery of All Nodes via Network Query
- 12.8 Discovery of Selected Node via Network Query

[Appendix A – Diagnostics via USB Console](#)

- A.1 Connecting to Console
- A.2 Commands

[Appendix B – LonWorks Trouble Shooting](#)

- B.1 General Practice, LED Indicators
- B.2 Errors when Sending Configuration
- B.3 Node Status Not "Ready"
- B.4 Cannot Discover Node

[Appendix C – BACnet Trouble Shooting](#)

- C.1 Observing BACnet Errors, LED Indicators
- C.2 BACnet Object Reliability Codes

[Appendix D - BACnet CSV Object List Format](#)

- D.1 Data Labels on Header Line
- D.2 Example CSV Files and Imports

[Appendix E - BACnet Codes](#)

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

[Appendix F – LonWorks CSV Network Variable List Format](#)

- F.1 Data Labels on Header Line
- F.2 Example CSV File and Import

[Appendix G - Configuration XML File Format](#)

- G.1 Configuration Files - Do Not Edit XML

[Appendix H - USB Driver Installation](#)

- H.1 Driver Installation

[Appendix J - Hardware Details](#)

- J.1 Service Button & USB Connection
- J.2 Front Panel LED Indicators
- J.3 Internal Diagnostic LED Indicators
- J.4 RS-485 Line Termination & Bias
- J.5 Server Module Init Jumper



User Guide

Babel Buster 2

**Models BB2-3020-NB, BB2-7020-NB
LonWorks BACnet Gateway
Rev. 1.0 – June 2015**

© 2015 Control Solutions, Inc.

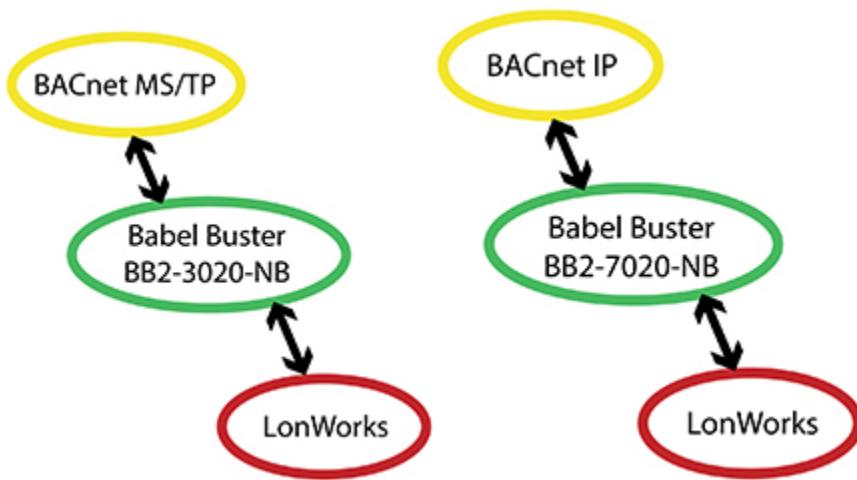
1 Introduction

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 Non-Bound Gateway Devices



What is different about BB2-3020-NB versus BB2-3020 (or BB2-7020-NB versus BB2-7020)? The LonMark certified BB2-3020 and BB2-7020 are designed for accessing BACnet devices from a LonWorks network. The "-NB" version, where NB stands for Non-Bound, is designed for accessing LonWorks devices from a BACnet network. Whereas the LonMark certified gateways require the use of a tool such as LonMaker to bind the gateways to the rest of the network, no such binding tool is required for the NB. The NB uses non-bound polling to access network variables in LonWorks devices and makes that data accessible as BACnet objects on the BACnet network. Device management in the NB is provided by the gateway along with the support of the configuration tool software provided at no charge by Control Solutions.

Babel Buster model BB2-3020-NB 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-NB 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-NB 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:

License agreement for SmartWin++ (BSD license)

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

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

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

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



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 in 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-NB and BB2-7020-NB 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.

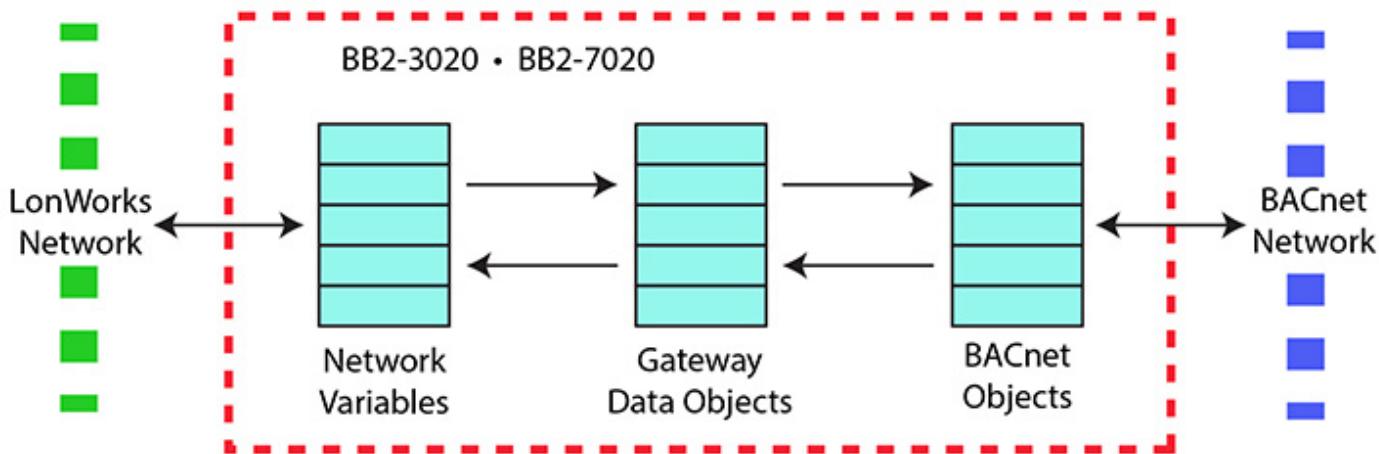
2.2 Data Flow in the Gateway

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 the devices. The NVO is sending "output" to the LonWorks network.

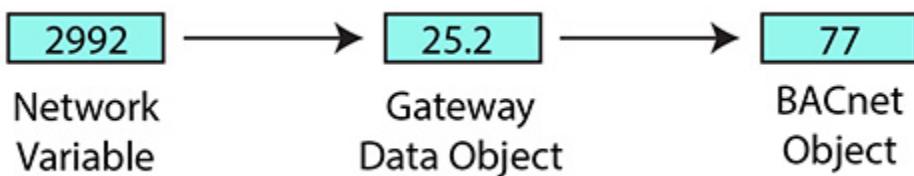
There are three different realizations of the same data within the LonWorks gateway. The direct connection to the LonWorks network is the Network Variable. The direct connection to the BACnet network is the BACnet object. The translating connection in between is a gateway data object.



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.

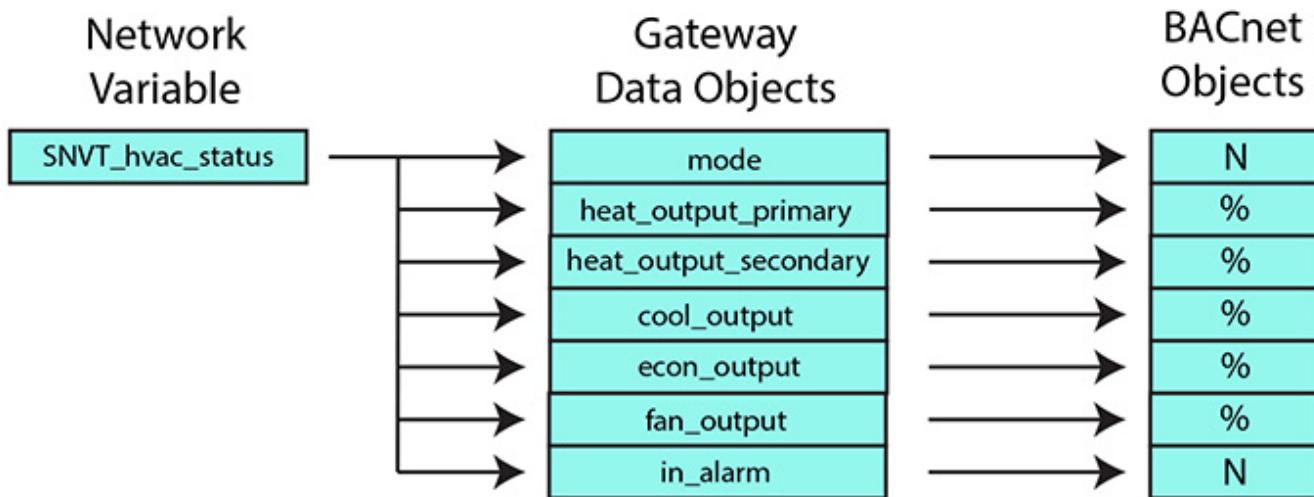


Next we will discuss an even more compelling reason why data cannot simply be sent "as is". Many LonWorks Network Variables are "structure", meaning a single network variable actually contains multiple pieces of data. A BACnet object can contain only a single value. Therefore, it is not possible to do 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 (download a copy of the LonMark Resource Files). 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 Non-Bound Polling of Network Variables

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 NB version of the Babel Buster 2 LonWorks Gateway has only a Node Object to allow it to exist peacefully on a LonWorks network. It does not contain any bindable network variables (other than Node Object variables) because it only does non-bound polling of network variables in other devices.

The polling of network variables in other devices is established by mappings in the gateway. Each of the gateway's internal data objects has a map telling it which LonWorks device to query, and which network variable in that device to access. Data is read from other devices using the network management "NV Fetch" request/response. Data is written to other devices using the network management "NV Update" request. Before the NB gateway can communicate with the other LonWorks device, it must establish the other device's network address, and this is done using the network management "Set Domain" request. There is more discussion related to device addressing in Section 12 and Appendix B.

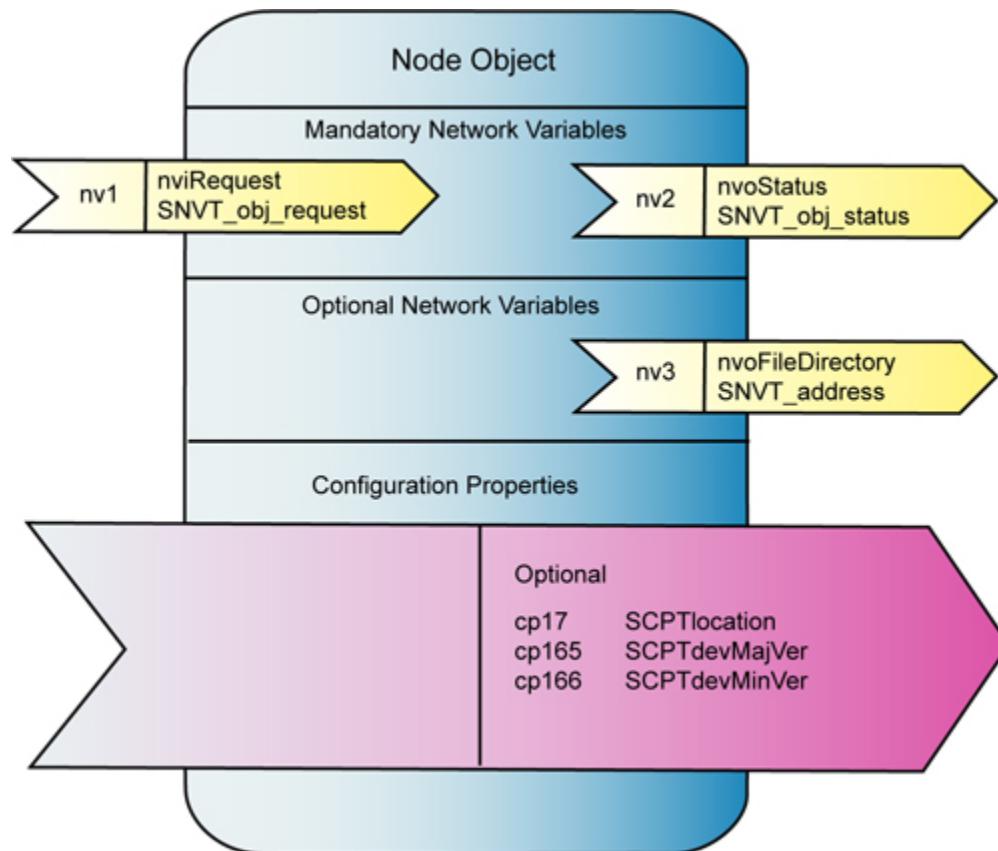
The BB2-3020-NB or BB2-7020-NB can poll up to 300 network variables from up to 50 other LonWorks devices.

Object Type	LON Object #	NV index
Node	0	0 .. 2

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 The Basics

The goal is to make one or more LonWorks devices accessible on a BACnet network, with the content of the LonWorks network variables showing up as BACnet objects. The configuration of the gateway as a BACnet device is done on the BACnet Port page. This is where you will assign a device instance, give it a presence on the network (MAC address, IP address, etc, as applicable), give the device object a name, and so on. Most importantly, the BACnet Port page is where you will allocate the available BACnet objects to the desired types, such as Analog Input, Analog Output, Binary Input, and so on. You are, in effect, creating a unique custom BACnet device.

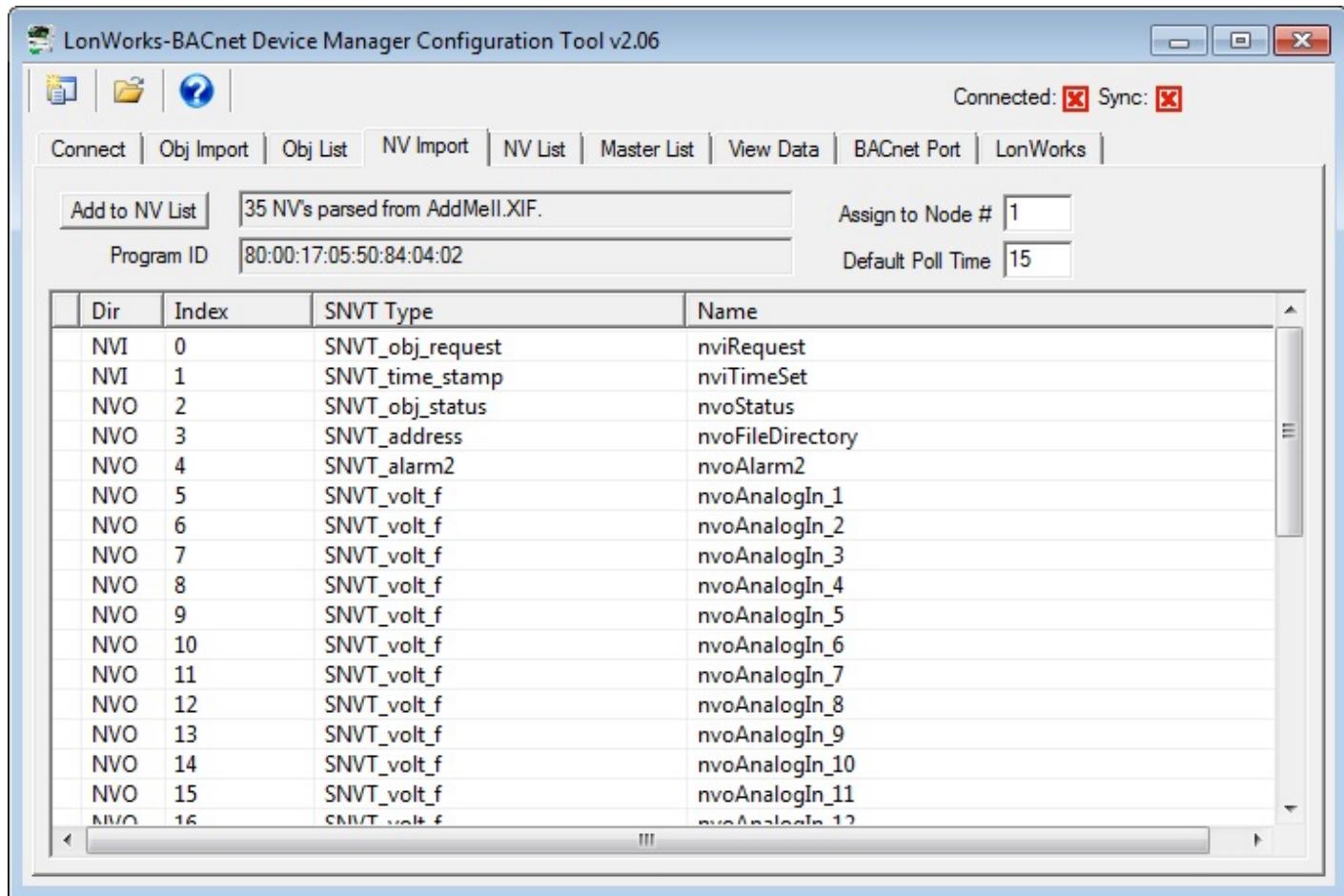
The Babel Buster gateway can be active or passive on the BACnet side, functioning as a client or server. As a server, the LonWorks data shows up in BACnet objects which some other BACnet device can then retrieve using Read Property, or write using Write Property (to Present Value property). If you are connecting the LonWorks device(s) to a BACnet BMS system via the Babel Buster gateway, then the BMS system will most likely be polling the gateway (or subscribing to COV notifications).

You have the option of making the gateway actively write data (received from LonWorks) to other BACnet devices, or read data from other BACnet devices and send it to the LonWorks device(s). If you want the gateway to do the polling, rather than BMS, then you will need to configure the BACnet client mappings to tell the gateway which BACnet devices to talk to, and which BACnet objects in those devices the gateway should interact with. If the Babel Buster gateway will be a BACnet slave (server), then you can skip all of the discussion pertaining to BACnet client mappings.

The topic you will spend most of your effort on is identifying the LonWorks network variables that you want to access from BACnet. Since some LonWorks network variables are structures with multiple data items per variable, you will sometimes need to map multiple BACnet objects (within the gateway) to a single LonWorks network variable. There are multiple possible starting points for configuring the gateway, and these are described in the following sections.

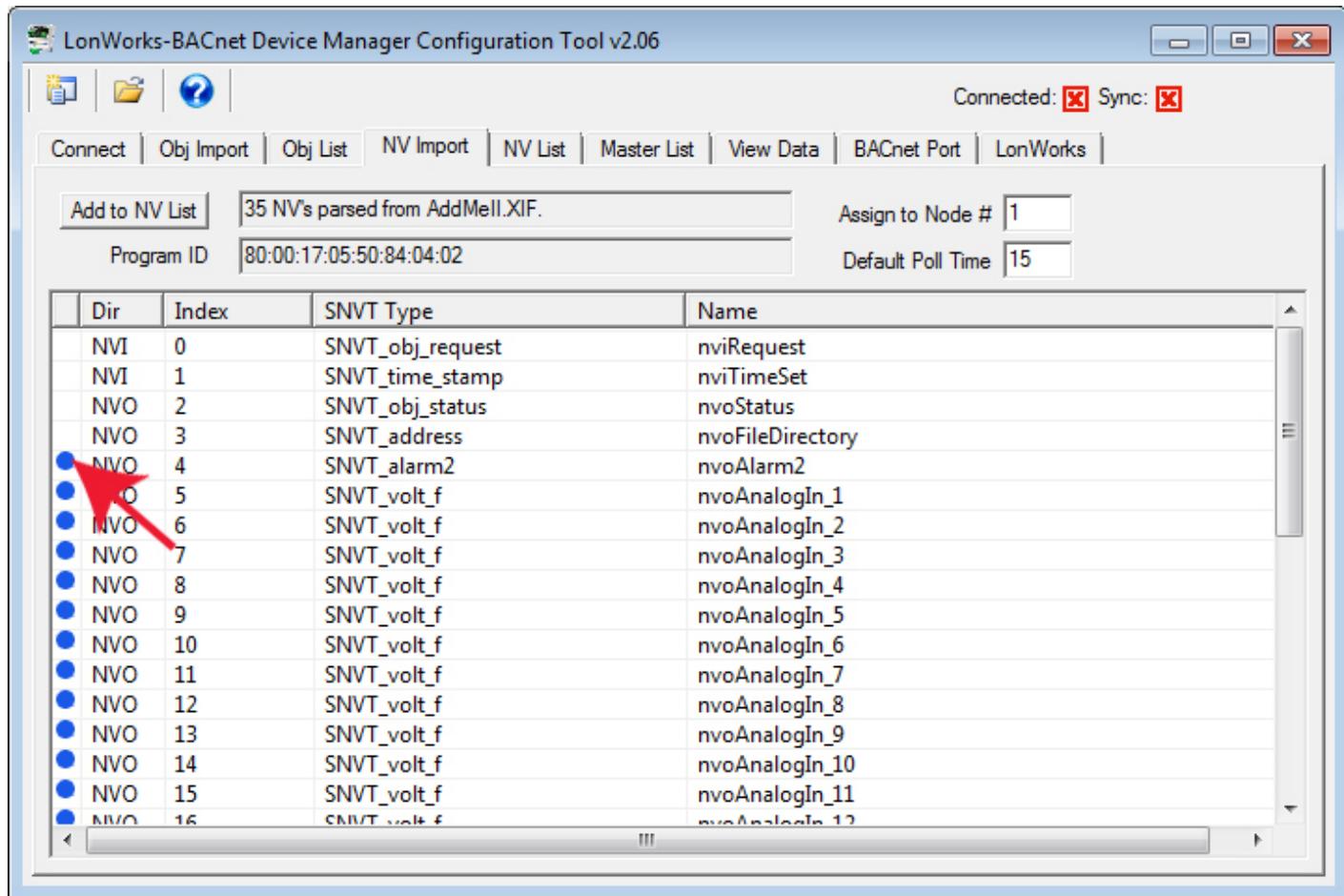
3.2 Build Configuration from XIF File for LonWorks Device

To begin building your configuration from an XIF file (obtained from the manufacturer of the device), 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.

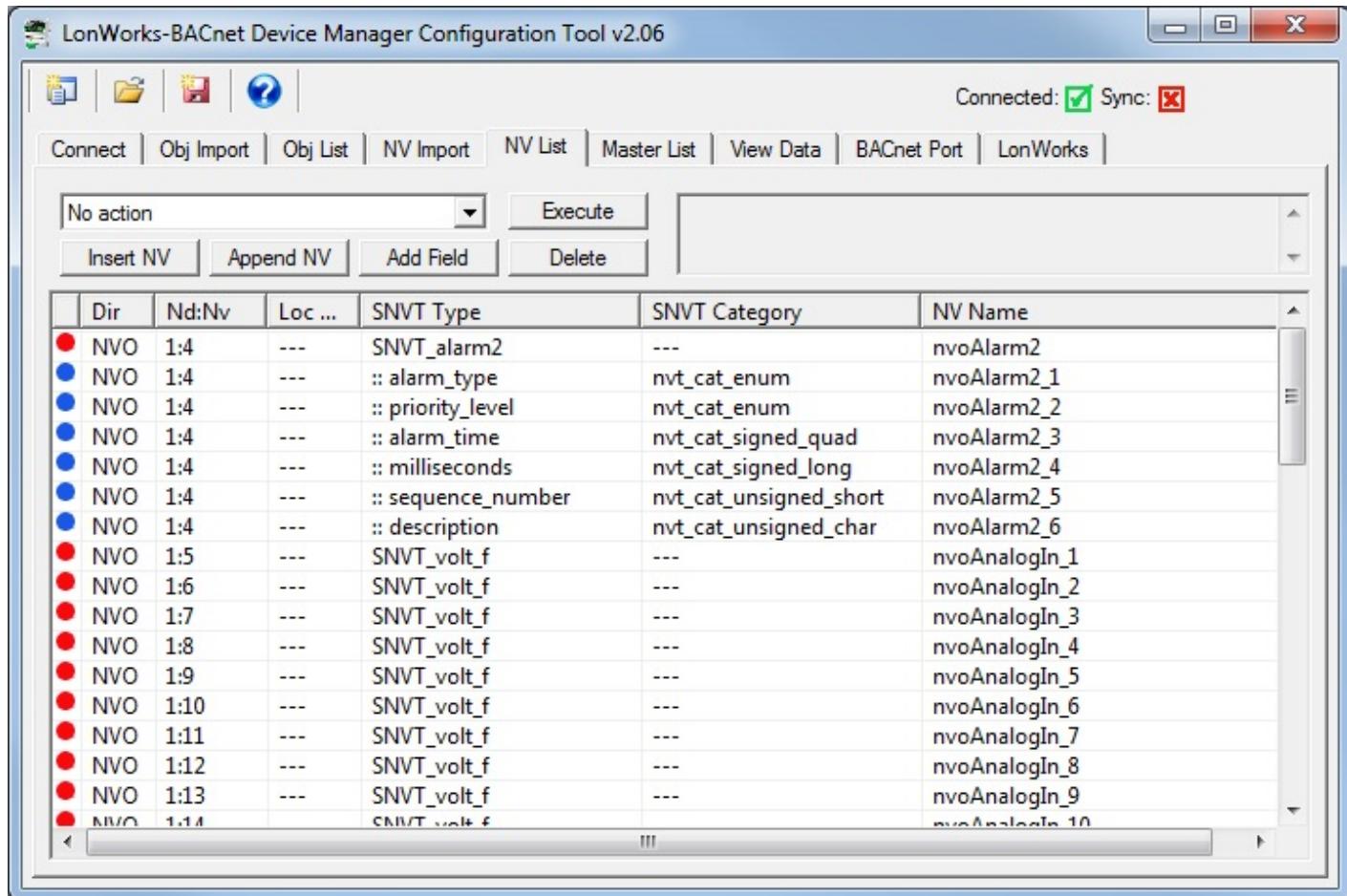


The NV Import page is essentially a scratch pad where you import the content of XIF files, then select which of the available variables you wish to include in your gateway configuration. Click on the icon column header to select all items, or click on the icon column for individual lines to select only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

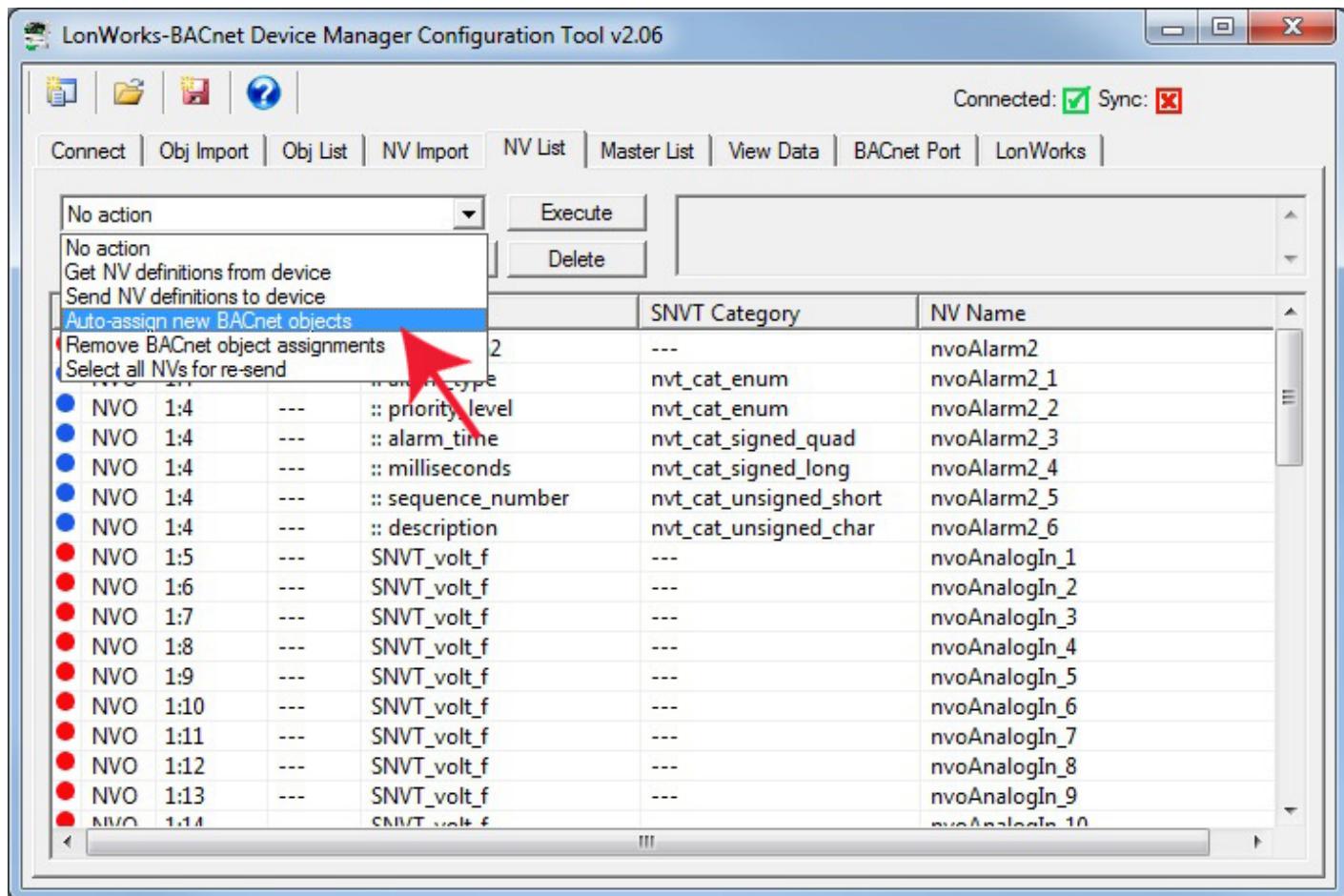
When you have made your selections, click Add to NV List. The selected variables are now copied to the NV List.



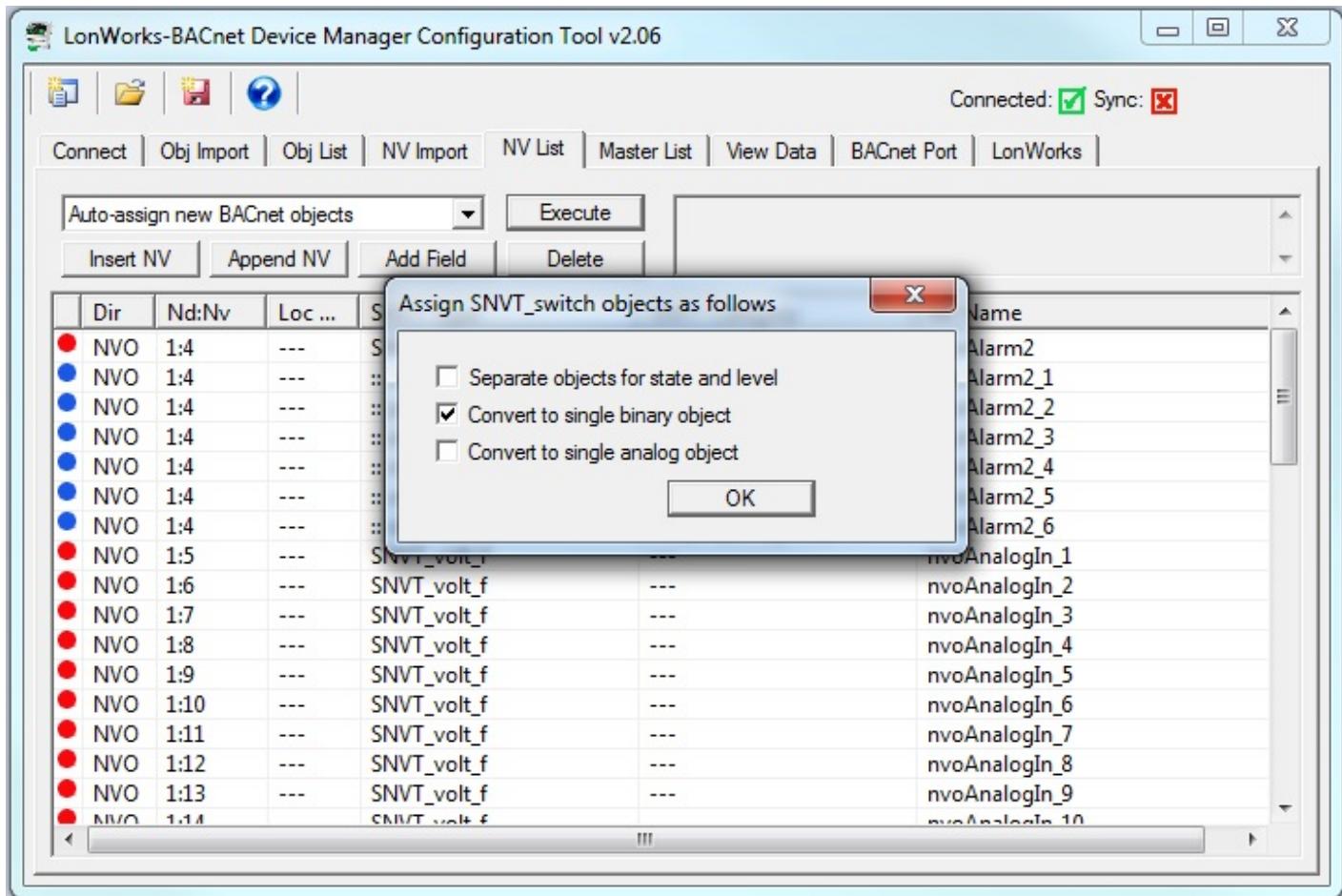
The NV List is the definition of the list of network variables found in other LonWorks devices that will be mapped to objects in your gateway. The icon in the first column will be red if this NV definition has not yet been written to the gateway, and green if it has been written to the gateway. Blue 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).



At this point, you have the option of editing the NV list and also the option of manually assigning variables to BACnet objects. But the easiest way to quickly configure the gateway is to simply select "Auto-assign new BACnet objects" from the list and click Execute. The tool will make a "best guess" on appropriate object type based on NV data type and direction.



When auto-assigning objects, a dialog will pop up asking about your preference for treatment of any SNVT_switch variables that may be in the list. LonWorks treatment of switches assumes a dimmer type switch, and the SNVT_switch variable has both a state and a level, both of which must be provided on the LonWorks side. You have 3 options for how you will treat this on the BACnet side: (a) You can assign two objects, one each for state and level; (b) You can assign an analog object which is then anticipating a value between 0 and 100 (percent implied); or (c) You can assign a binary object, in which case the gateway automatically translates 'inactive' into the 'off' state with 0% level, and 'active' into the 'on' state with 100% level.



After object assignment, the Local Object column will be populated. The example below shows several network variables that will be polled in node #1 in the node table on the LonWorks page of the tool. The Nd:Nv column shows node number in that table, and NV index that will be queried in that node. The NV index value is found in the XIF file. If creating the NV list by other means, you will need to find out from the device manufacturer's documentation what the NV index is for the variables of interest. This is effectively the "address" of the variable in the LonWorks device.

The "Dir" column shows NVO for Network Variable Output, meaning the LonWorks device will transmit data to the LonWorks network, or NVO for Network Variable Input, meaning the LonWorks device expects to receive data from the LonWorks network.

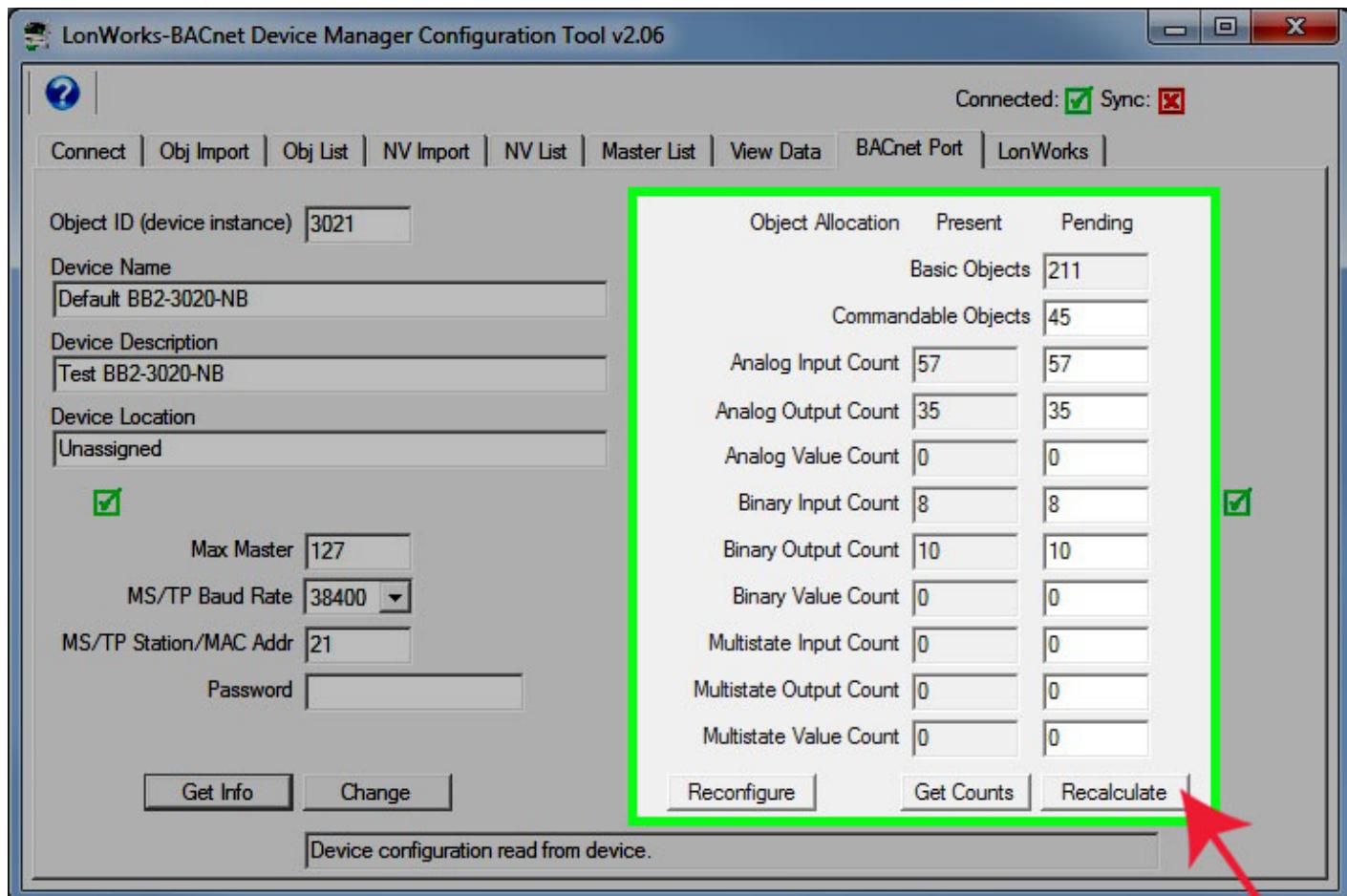
Looking at nvoAnalogIn_1 as an example, the LonWorks device at node #1 will be providing a floating point data value representing voltage. The Babel Buster gateway will be polling this variable to read its data from the LonWorks device. Your BACnet client (e.g. BMS front end) can then read the Present Value of Analog Input instance 7 from the gateway (at the gateway's device instance found on the BACnet Port page) and see that voltage data.

Dir	Nd:Nv	Loc ...	SNVT Type	SNVT Category	NV Name
• NVO	1:4	---	SNVT_alarm2	---	nvoAlarm2
• NVO	1:4	AI1	:: alarm_type	nvt_cat_enum	nvoAlarm2_1
• NVO	1:4	AI2	:: priority_level	nvt_cat_enum	nvoAlarm2_2
• NVO	1:4	AI3	:: alarm_time	nvt_cat_signed_quad	nvoAlarm2_3
• NVO	1:4	AI4	:: milliseconds	nvt_cat_signed_long	nvoAlarm2_4
• NVO	1:4	AI5	:: sequence_number	nvt_cat_unsigned_short	nvoAlarm2_5
• NVO	1:4	AI6	:: description	nvt_cat_unsigned_char	nvoAlarm2_6
• NVO	1:5	AI7	SNVT_volt_f	---	nvoAnalogIn_1
• NVO	1:6	AI8	SNVT_volt_f	---	nvoAnalogIn_2
• NVO	1:7	AI9	SNVT_volt_f	---	nvoAnalogIn_3
• NVO	1:8	AI10	SNVT_volt_f	---	nvoAnalogIn_4
• NVO	1:9	AI11	SNVT_volt_f	---	nvoAnalogIn_5
• NVO	1:10	AI12	SNVT_volt_f	---	nvoAnalogIn_6
• NVO	1:11	AI13	SNVT_volt_f	---	nvoAnalogIn_7
• NVO	1:12	AI14	SNVT_volt_f	---	nvoAnalogIn_8
• NVO	1:13	AI15	SNVT_volt_f	---	nvoAnalogIn_9
• NVO	1:14	AI16	SNVT_volt_f	---	nvoAnalogIn_10

There is one more bit of housekeeping that must be done before your gateway configuration can be sent from the tool to the Babel Buster gateway device. You must first allocate the counts of BACnet objects that will appear in the gateway. Go to the BACnet Port page to do this. If you have just recently completed an auto-assignment of BACnet objects, you can click the Recalculate button to get an updated count of what objects are going to be required by your configuration. Recalculate, and then Reconfigure.

It is important to plan your object counts as far in advance as possible. Once you have started sending configuration information to the gateway, any reallocation of object counts will result in that configuration being lost. If you have a configuration that you want to save while you reconfigure the object counts, go to the Master List page and click the file save icon at the top. Saving your configuration is a good idea just in case you accidentally close the program. Reconfiguring the gateway device will erase configuration in the device, but not clear configuration currently retained within the tool program. You can reconfigure, and then resend configuration to restore gateway device configuration without reloading any files.

The other thing that must be done is to connect with the LonWorks device identified by node number in the NV List. This is done from the LonWorks page, and the procedure for making this connection is covered in that section of this user guide.

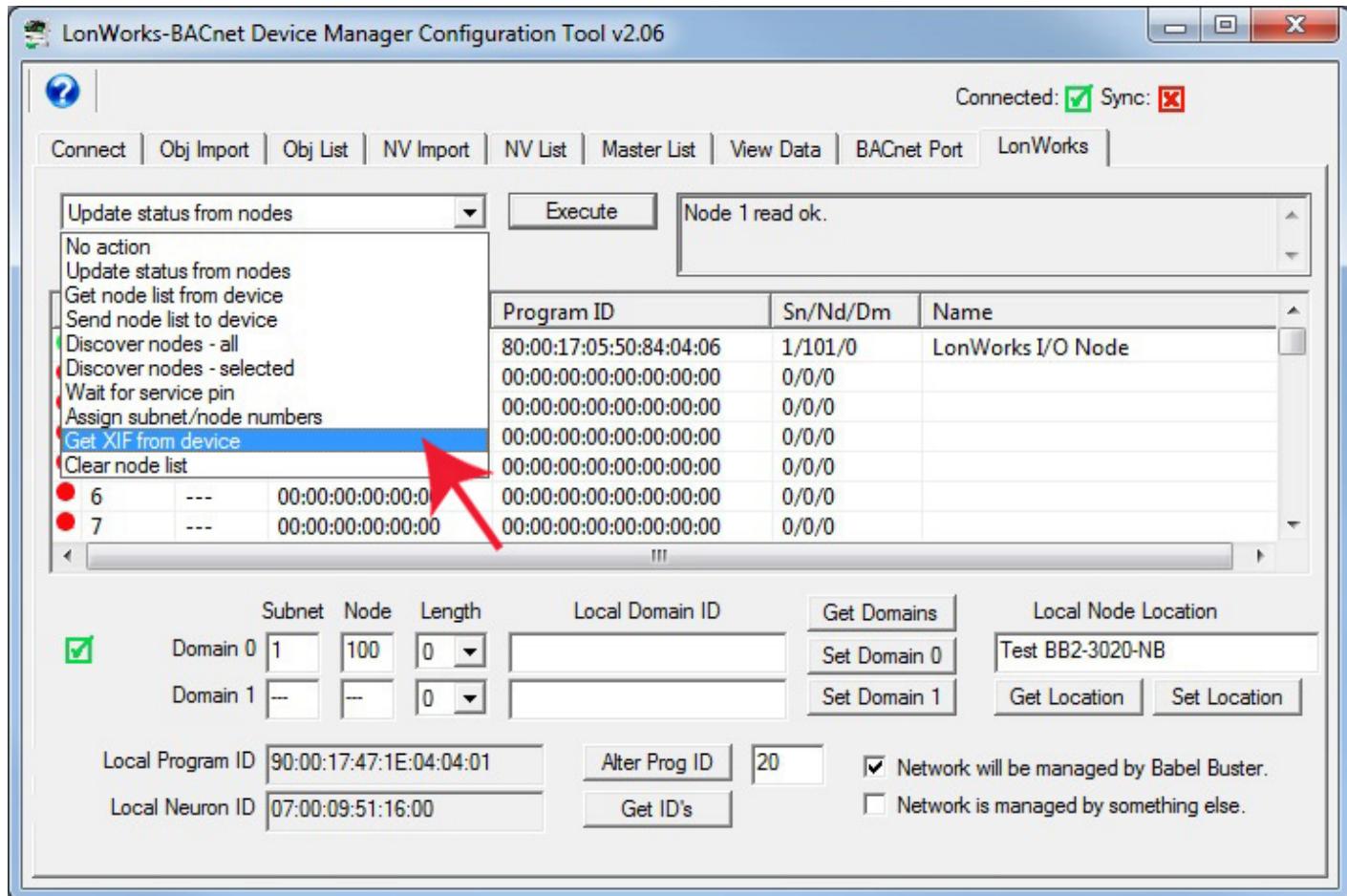


3.3 Build Configuration by Importing XIF File from LonWorks Device

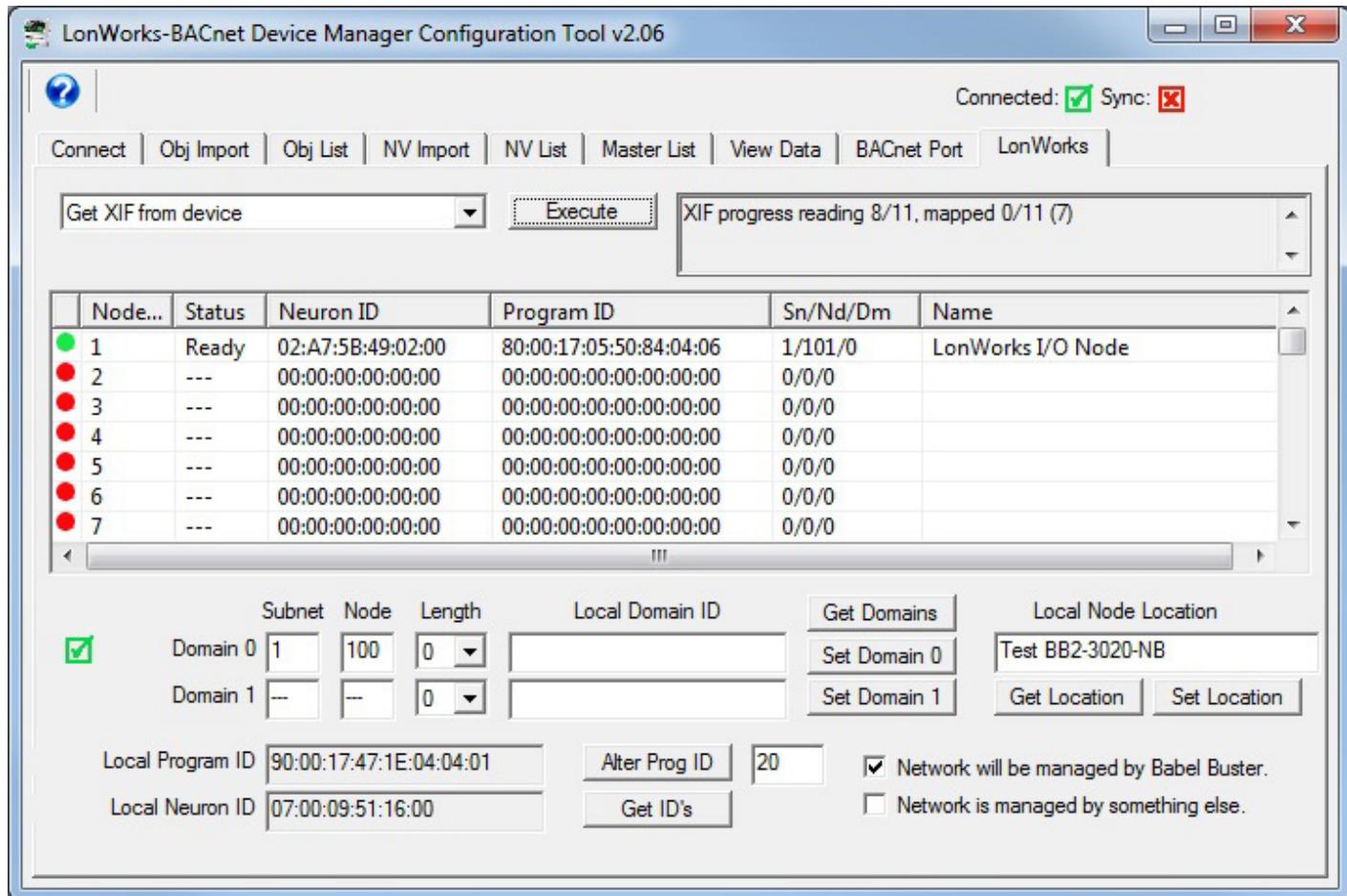
You can usually import the contents of an XIF file from the device itself in the event you do not have ready access to the XIF file. If the device contains more than 300 network variables, then importing from the device will exceed the capacity of the gateway. Also, it is known that some devices block import of the XIF file from the device itself. Therefore, while it is worth a try and will succeed in the majority of cases, there is no guarantee.

You will also need to first connect with the node using the procedures outlined for the LonWorks page. If you cannot communicate with the node, you cannot import its XIF information. Therefore, you must establish communication first (whereas if you have the XIF file, you do not need to be connected to the device at all to initially create a gateway configuration).

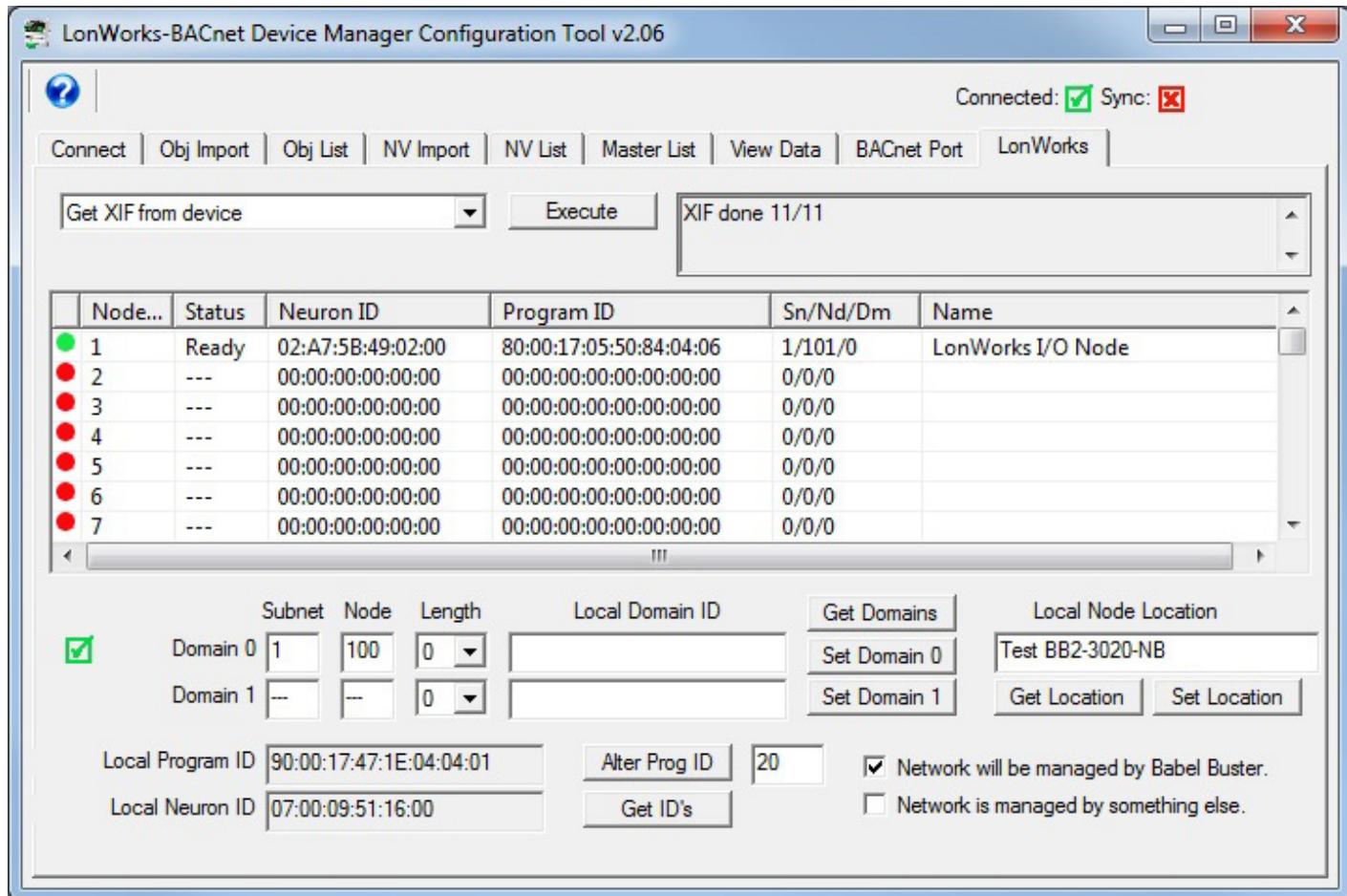
Assuming you have already connected with the node to be imported, and its status is 'Ready', select 'Get XIF from device' from the list on the LonWorks page and click Execute.



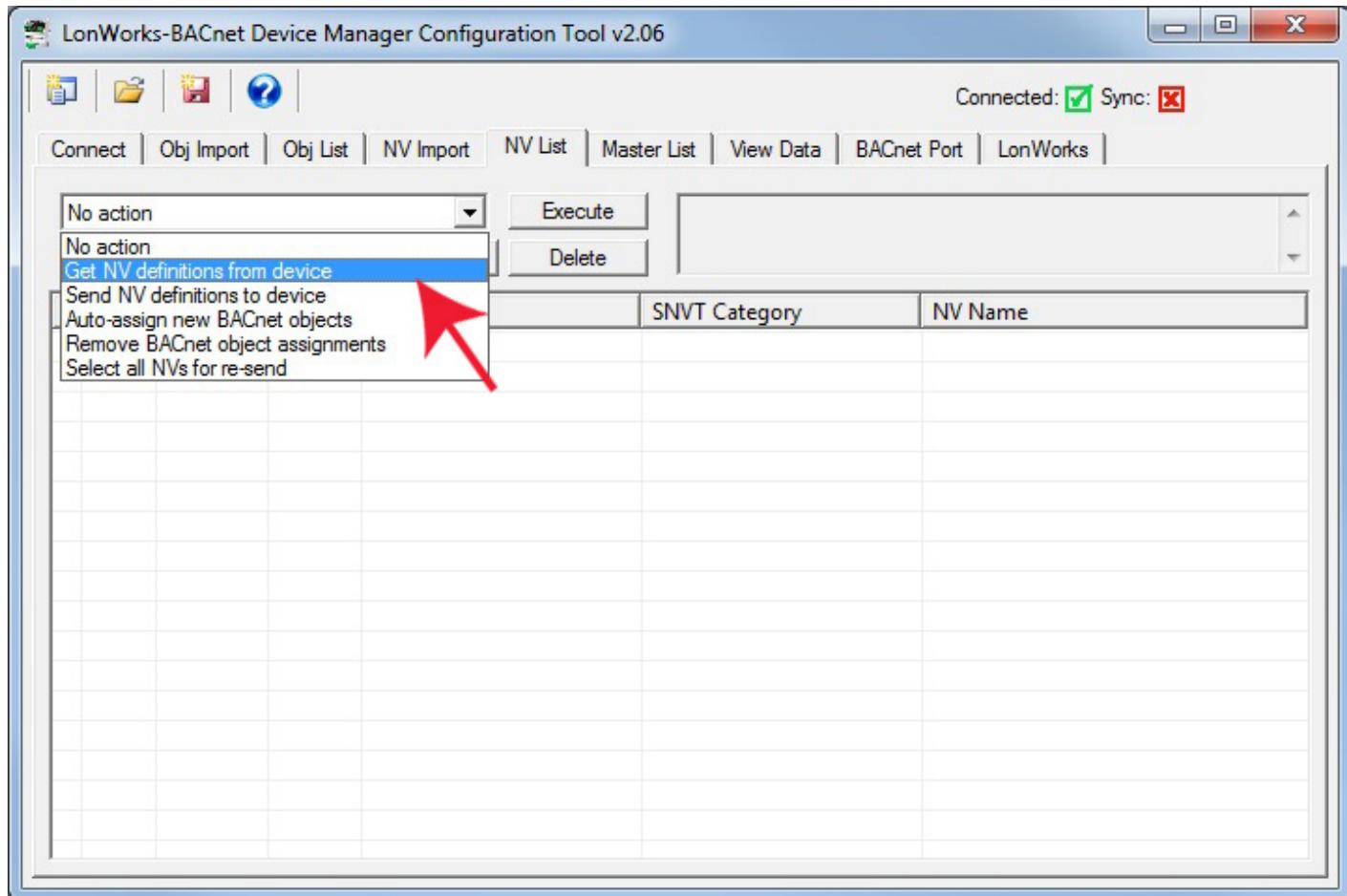
During the XIF import process, progress will be updated in the tool status window from time to time. The process usually involves multiple passes, so do not be alarmed if progress counts revert to zero and begin counting up again. In most cases, three passes are required.



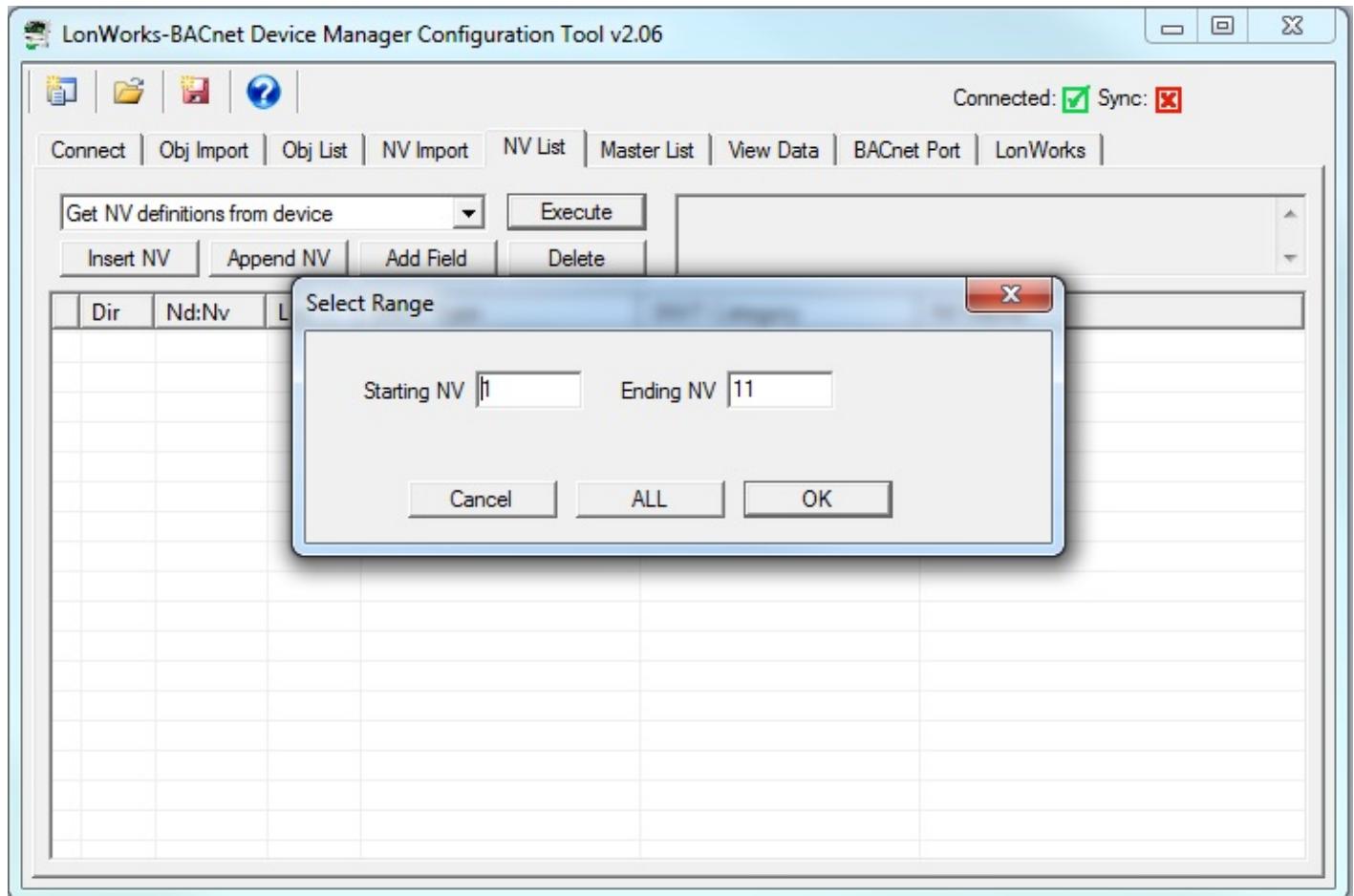
Once complete, the message "XIF done" will appear, along with an indication of how many Network Variables were imported.



The XIF import process filled up the NV table in the gateway device. To work with this information, you now need to retrieve the NV List from the gateway device. Select 'Get NV definitions from device' and click Execute.



Upon executing 'Get NV definitions from device', a dialog will pop up asking for the range of network variables to retrieve. It will default to all of those found in the device. Usually you can just click OK here. If you have imported more than one device, and only want to retrieve information from the most recent import, then assuming you kept track of previous counts, you can enter a range less than all available.



Once the process of retrieving the imported NV List is complete, the process is largely the same as if you had read the XIF from a file.

The screenshot shows the LonWorks-BACnet Device Manager Configuration Tool v2.06. The window title is "LonWorks-BACnet Device Manager Configuration Tool v2.06". The menu bar includes "File", "Edit", "View", "Tools", "Help", and "About". The toolbar contains icons for "Connect", "Obj Import", "Obj List", "NV Import", "NV List" (selected), "Master List", "View Data", "BACnet Port", and "LonWorks". Status indicators show "Connected: Sync: ". The main area has tabs for "Get NV definitions from device" (selected), "Execute", and a message box displaying "NV 10 read ok." and "NV 11 read ok.". Below are buttons for "Insert NV", "Append NV", "Add Field", and "Delete". A large table lists Network Variables (NVs) with columns: Dir, Nd:Nv, Loc ..., SNVT Type, SNVT Category, and NV Name. The table contains 20 rows of data.

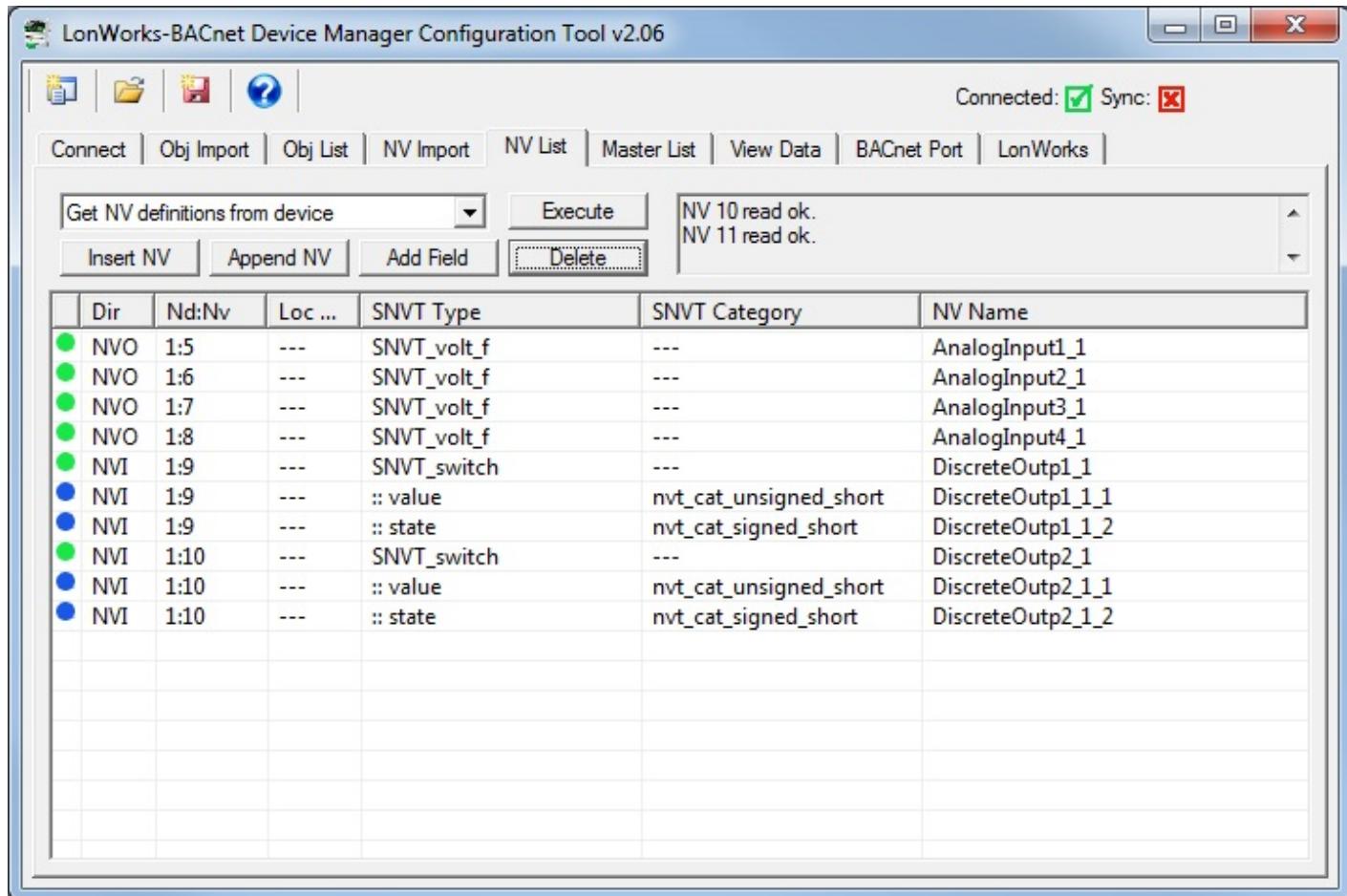
Dir	Nd:Nv	Loc ...	SNVT Type	SNVT Category	NV Name
● NVI	1:0	---	SNVT_obj_request	---	NodeObject1_1
● NVI	1:1	---	SNVT_time_stamp	---	NodeObject1_3
● NVI	1:1	---	:: year	nvt_cat_unsigned_long	NodeObject1_3_1
● NVI	1:1	---	:: month	nvt_cat_unsigned_short	NodeObject1_3_2
● NVI	1:1	---	:: day	nvt_cat_unsigned_short	NodeObject1_3_3
● NVI	1:1	---	:: hour	nvt_cat_unsigned_short	NodeObject1_3_4
● NVI	1:1	---	:: minute	nvt_cat_unsigned_short	NodeObject1_3_5
● NVI	1:1	---	:: second	nvt_cat_unsigned_short	NodeObject1_3_6
● NVO	1:2	---	SNVT_obj_status	---	NodeObject1_2
● NVO	1:3	---	SNVT_address	---	NodeObject1_8
● NVO	1:4	---	SNVT_alarm2	---	NodeObject1_9
● NVO	1:4	---	:: alarm_type	nvt_cat_enum	NodeObject1_9_1
● NVO	1:4	---	:: priority_level	nvt_cat_enum	NodeObject1_9_2
● NVO	1:4	---	:: alarm_time	nvt_cat_signed_quad	NodeObject1_9_3
● NVO	1:4	---	:: milliseconds	nvt_cat_signed_long	NodeObject1_9_4
● NVO	1:4	---	:: sequence_number	nvt_cat_unsigned_short	NodeObject1_9_5
● NVO	1:4	---	:: description	nvt_cat_unsigned_short	NodeObject1_9_6

The device may contain more Network Variables than you care to process as BACnet objects. You can select those variables and click Delete to remove them from your configuration.

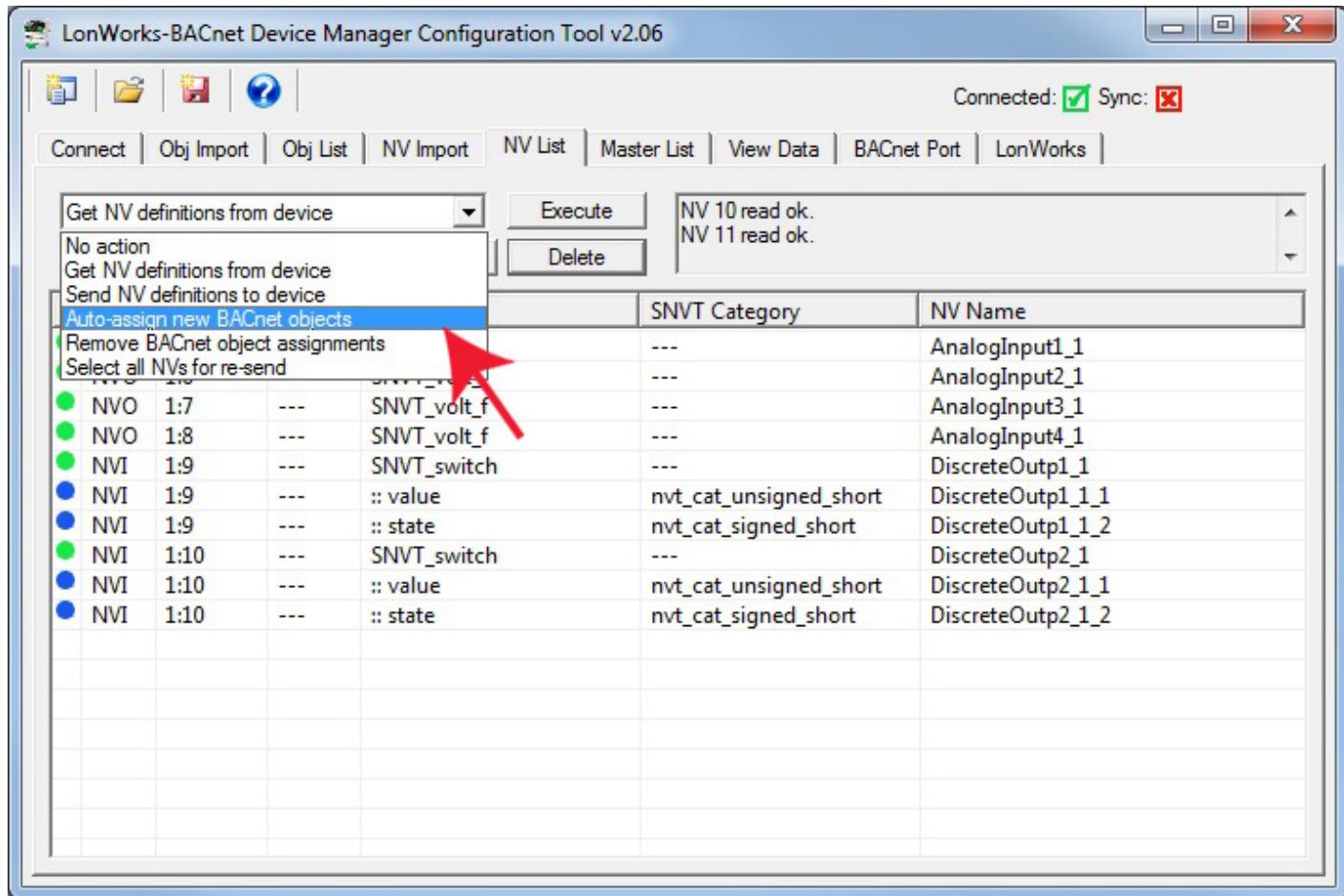
The screenshot shows the LonWorks-BACnet Device Manager Configuration Tool interface. The title bar reads "LonWorks-BACnet Device Manager Configuration Tool v2.06". The menu bar includes "Connect", "Obj Import", "Obj List", "NV Import", "NV List" (which is selected), "Master List", "View Data", "BACnet Port", and "LonWorks". The toolbar has icons for file operations and help. The status bar indicates "Connected: Sync: ". The main area has a message box saying "Get NV definitions from device" and "Execute" button, followed by "NV 10 read ok." and "NV 11 read ok.". Below is a table with columns: Dir, Nd:Nv, Loc ..., SNVT Type, SNVT Category, and NV Name. The table lists various NV entries, many of which have been deleted, as indicated by the red arrow pointing to the "Delete" button in the toolbar.

Dir	Nd:Nv	Loc ...	SNVT Type	SNVT Category	NV Name
NVI	1:1	---	SNVT_time_stamp	---	NodeObject1_3
NVI	1:1	---	:: year	nvt_cat_unsigned_long	NodeObject1_3_1
NVI	1:1	---	:: month	nvt_cat_unsigned_short	NodeObject1_3_2
NVI	1:1	---	:: day	nvt_cat_unsigned_short	NodeObject1_3_3
NVI	1:1	---	:: hour	nvt_cat_unsigned_short	NodeObject1_3_4
NVI	1:1	---	:: minute	nvt_cat_unsigned_short	NodeObject1_3_5
NVI	1:1	---	:: second	nvt_cat_unsigned_short	NodeObject1_3_6
NVO	1:2	---	SNVT_obj_status	---	NodeObject1_2
NVO	1:3	---	SNVT_address	---	NodeObject1_8
NVO	1:4	---	SNVT_alarm2	---	NodeObject1_9
NVO	1:4	---	:: alarm_type	nvt_cat_enum	NodeObject1_9_1
NVO	1:4	---	:: priority_level	nvt_cat_enum	NodeObject1_9_2
NVO	1:4	---	:: alarm_time	nvt_cat_signed_quad	NodeObject1_9_3
NVO	1:4	---	:: milliseconds	nvt_cat_signed_long	NodeObject1_9_4
NVO	1:4	---	:: sequence_number	nvt_cat_unsigned_short	NodeObject1_9_5
NVO	1:4	---	:: description	nvt_cat_unsigned_char	NodeObject1_9_6
NVO	1:5	---	SNVT_VOLT_F	---	AnalogInput1_1

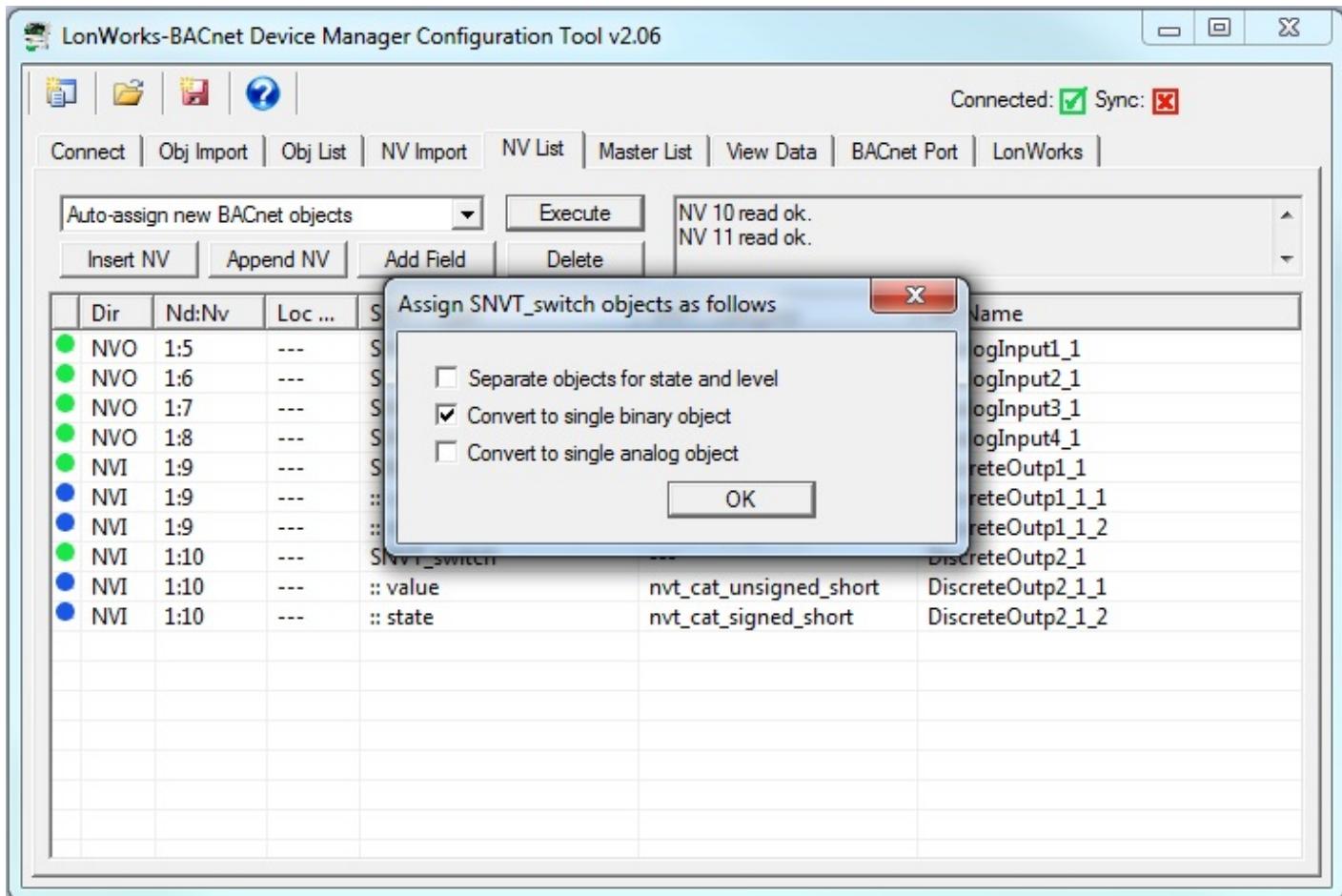
In this example, we have eliminated variables we don't care about, such as node object status (SNVT_obj_status). The list that remains is the actual I/O point list in this particular device.



At this point, you have the option of manually assigning variables to BACnet objects. But the easiest way to quickly configure the gateway is to simply select "Auto-assign new BACnet objects" from the list and click Execute. The tool will make a "best guess" on appropriate object type based on NV data type and direction.



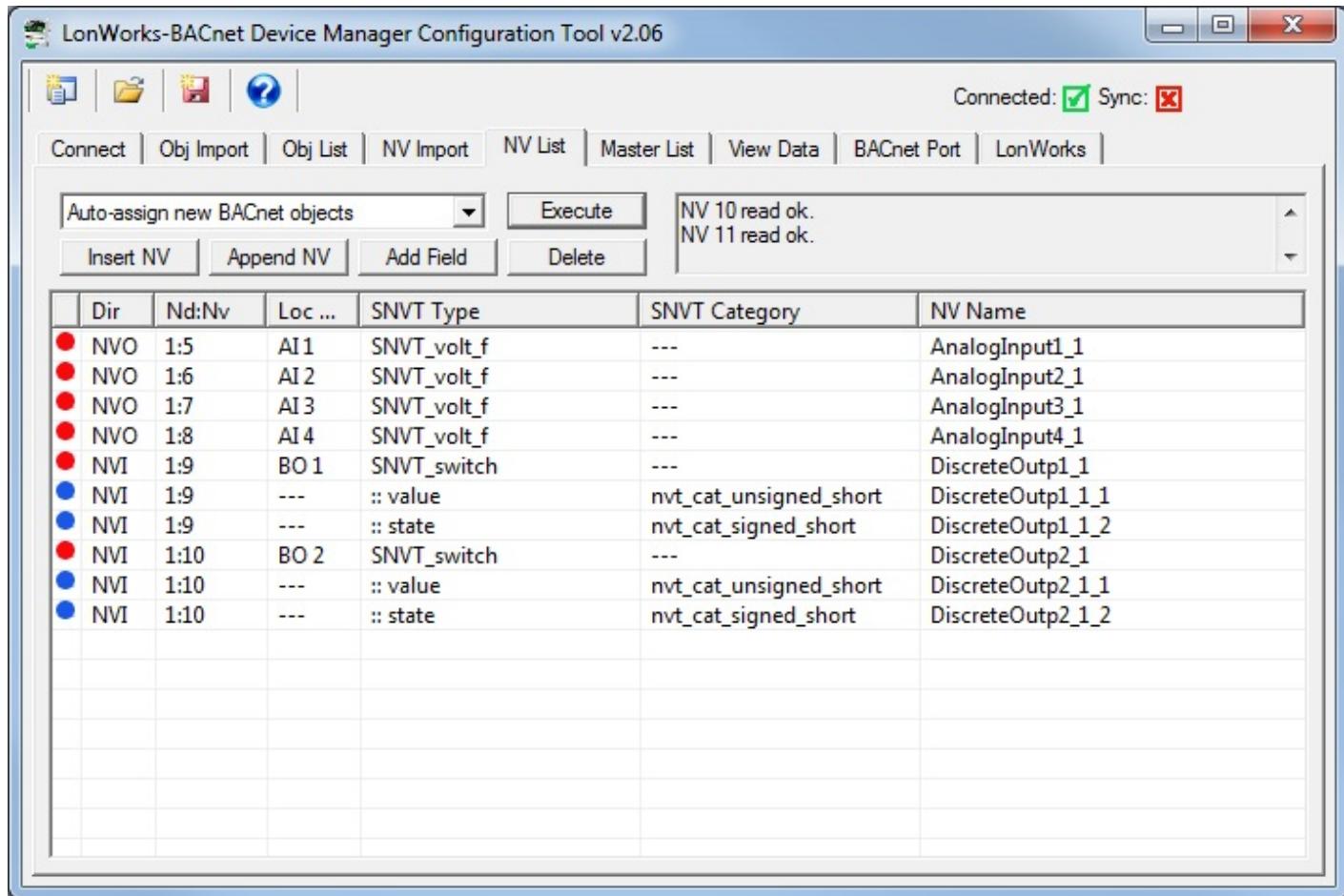
When auto-assigning objects, a dialog will pop up asking about your preference for treatment of any SNVT_switch variables that may be in the list. LonWorks treatment of switches assumes a dimmer type switch, and the SNVT_switch variable has both a state and a level, both of which must be provided on the LonWorks side. You have 3 options for how you will treat this on the BACnet side: (a) You can assign two objects, one each for state and level; (b) You can assign an analog object which is then anticipating a value between 0 and 100 (percent implied); or (c) You can assign a binary object, in which case the gateway automatically translates 'inactive' into the 'off' state with 0% level, and 'active' into the 'on' state with 100% level.



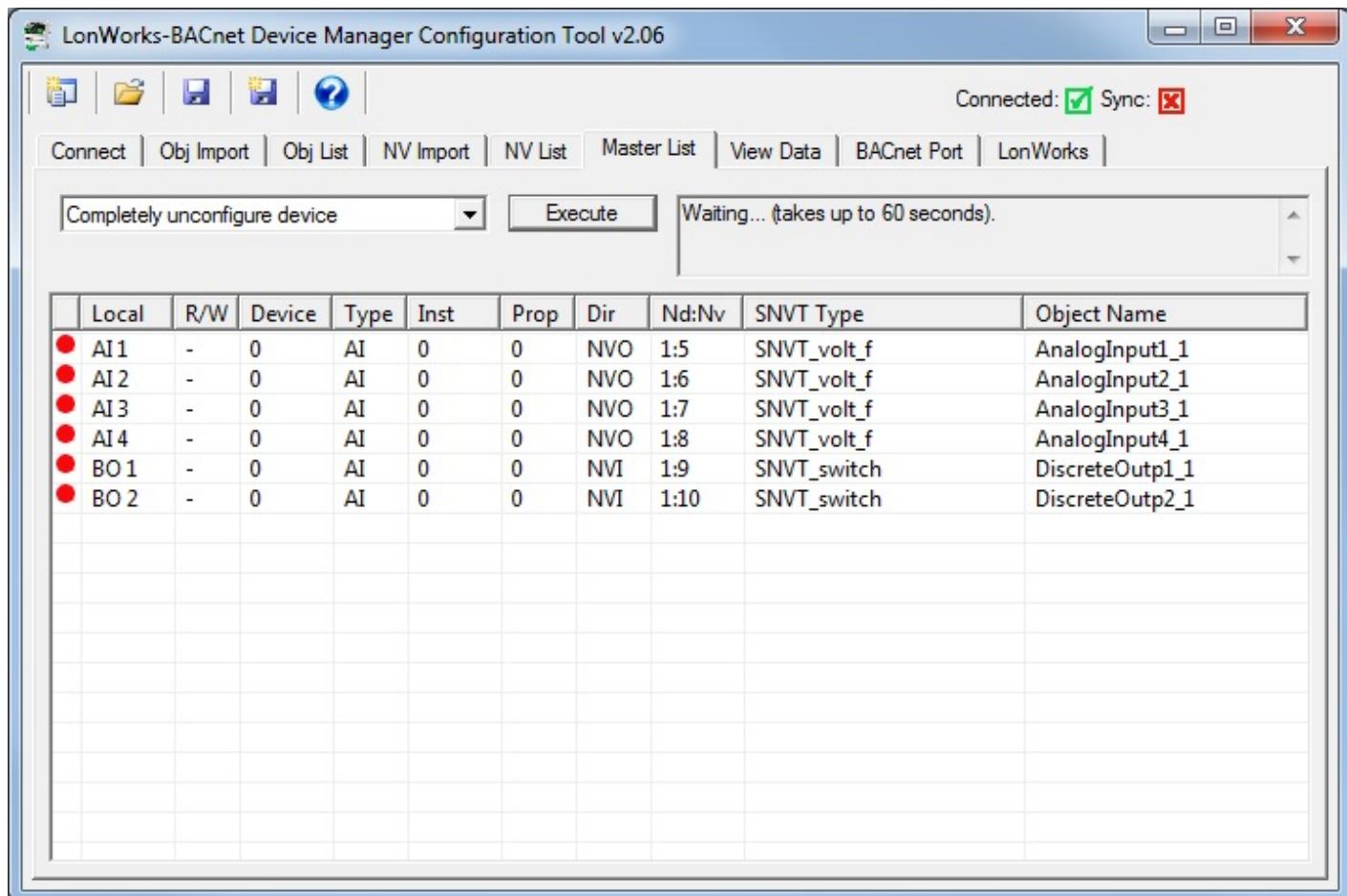
After object assignment, the Local Object column will be populated. The example below shows several network variables that will be polled in node #1 in the node table on the LonWorks page of the tool. The Nd:Nv column shows node number in that table, and NV index that will be queried in that node. The NV index value is found in the XIF file. If creating the NV list by other means, you will need to find out from the device manufacturer's documentation what the NV index is for the variables of interest. This is effectively the "address" of the variable in the LonWorks device.

The "Dir" column shows NVO for Network Variable Output, meaning the LonWorks device will transmit data to the LonWorks network, or NVO for Network Variable Input, meaning the LonWorks device expects to receive data from the LonWorks network.

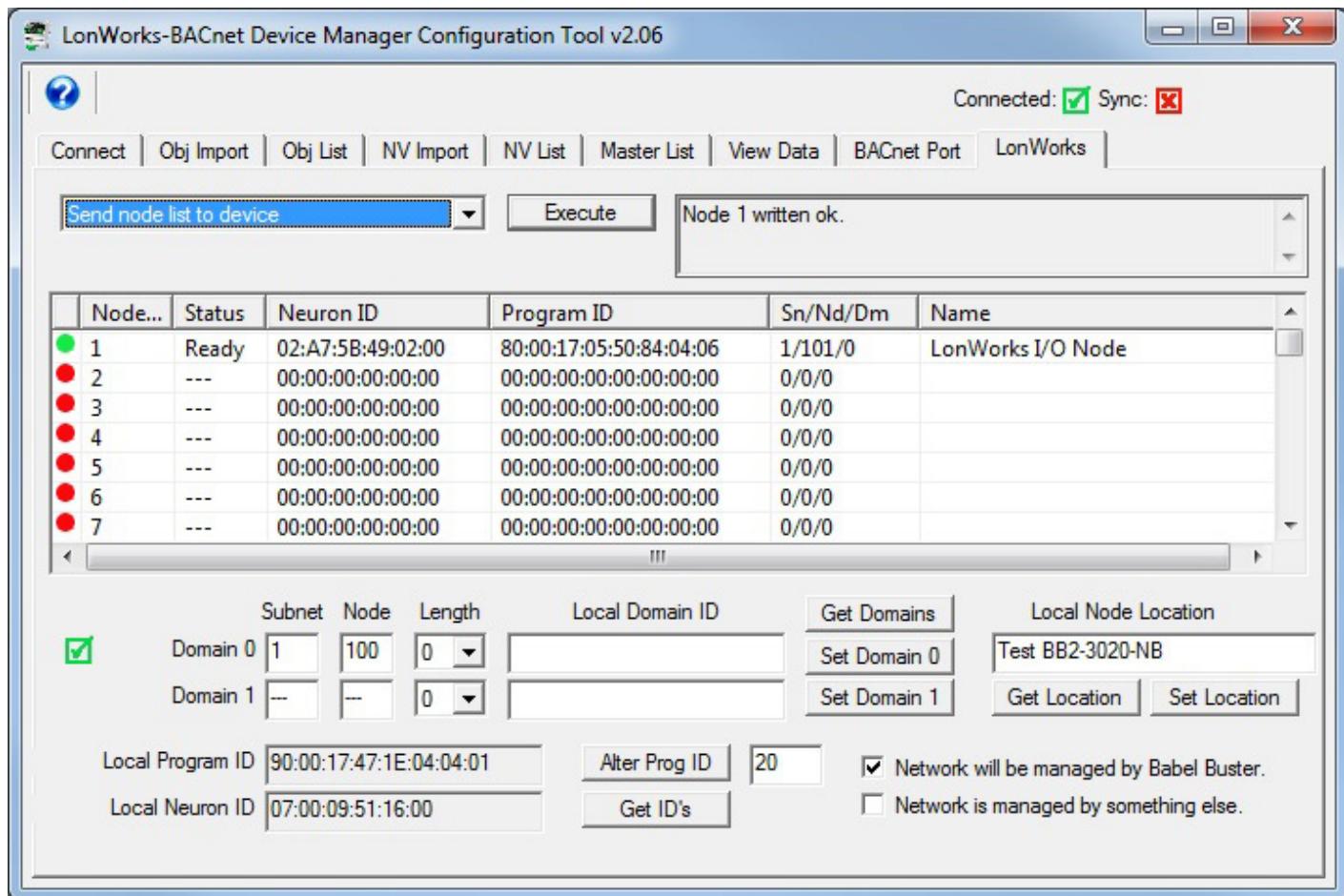
Looking at nvoAnalogIn_1 as an example, the LonWorks device at node #1 will be providing a floating point data value representing voltage. The Babel Buster gateway will be polling this variable to read its data from the LonWorks device. Your BACnet client (e.g. BMS front end) can then read the Present Value of Analog Input instance 1 from the gateway (at the gateway's device instance found on the BACnet Port page) and see that voltage data.



There is one IMPORTANT bit of housekeeping to do here before sending your revised NV List to the gateway device. The gateway itself still contains the originally imported list of variables. You don't want those. Therefore, select 'Completely unconfigure device' from the list and click Execute. This is the quicker way to clear the list. The other option is to select ALL when sending the NV List to the gateway device. Sending all 300 NV entries will take some time, but will make sure that the NV List in the device matches what you see in the configuration tool.



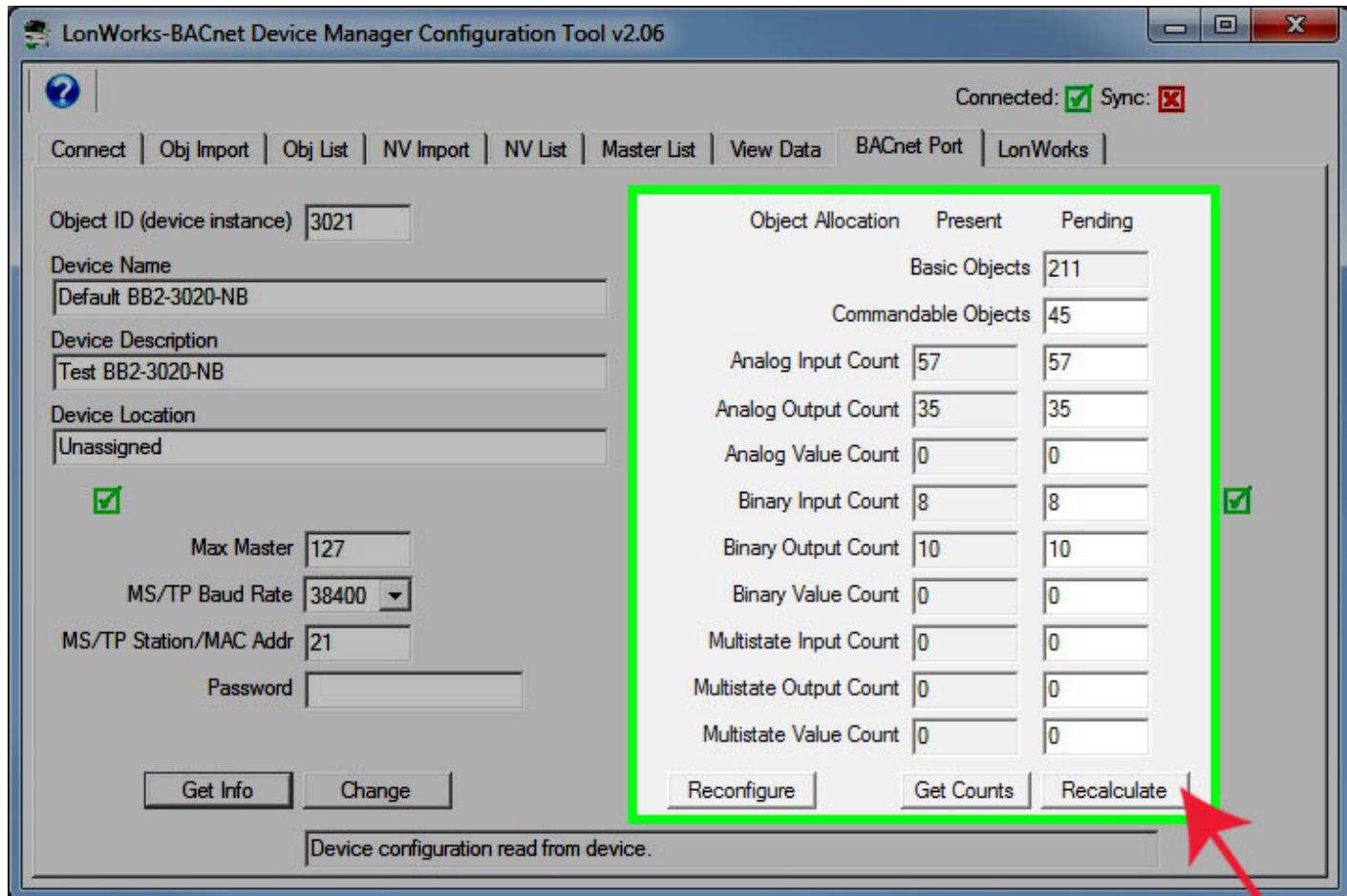
Assuming you did the 'Completely unconfigure device', you now need to restore the node table. You need to either keep the tool open and resend the list retained by the tool, or reload the configuration file you previously saved on the Master List page. Resend the node list by selecting 'Send node list to device' and click Execute.



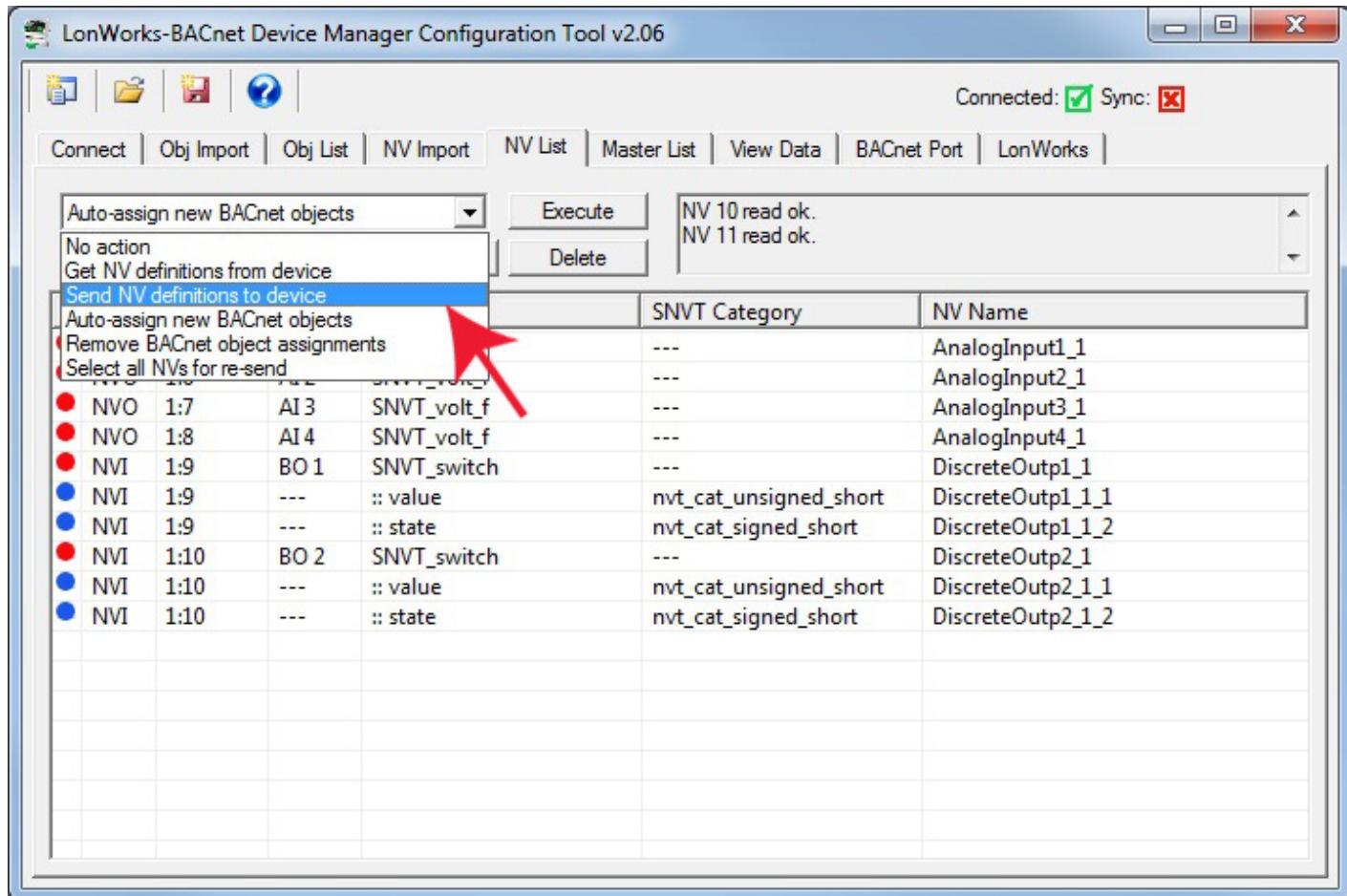
There is one more bit of housekeeping that must be done before your gateway configuration can be sent from the tool to the Babel Buster gateway device. You must first allocate the counts of BACnet objects that will appear in the gateway. Go to the BACnet Port page to do this. If you have just recently completed an auto-assignment of BACnet objects, you can click the Recalculate button to get an updated count of what objects are going to be required by your configuration. Recalculate, and then Reconfigure.

It is important to plan your object counts as far in advance as possible. Once you have started sending configuration information to the gateway, any reallocation of object counts will result in that configuration being lost. If you have a configuration that you want to save while you reconfigure the object counts, go to the Master List page and click the file save icon at the top. Saving your configuration is a good idea just in case you accidentally close the program. Reconfiguring the gateway device will erase configuration in the device, but not clear configuration currently retained within the tool program. You can reconfigure, and then resend configuration to restore gateway device configuration without reloading any files.

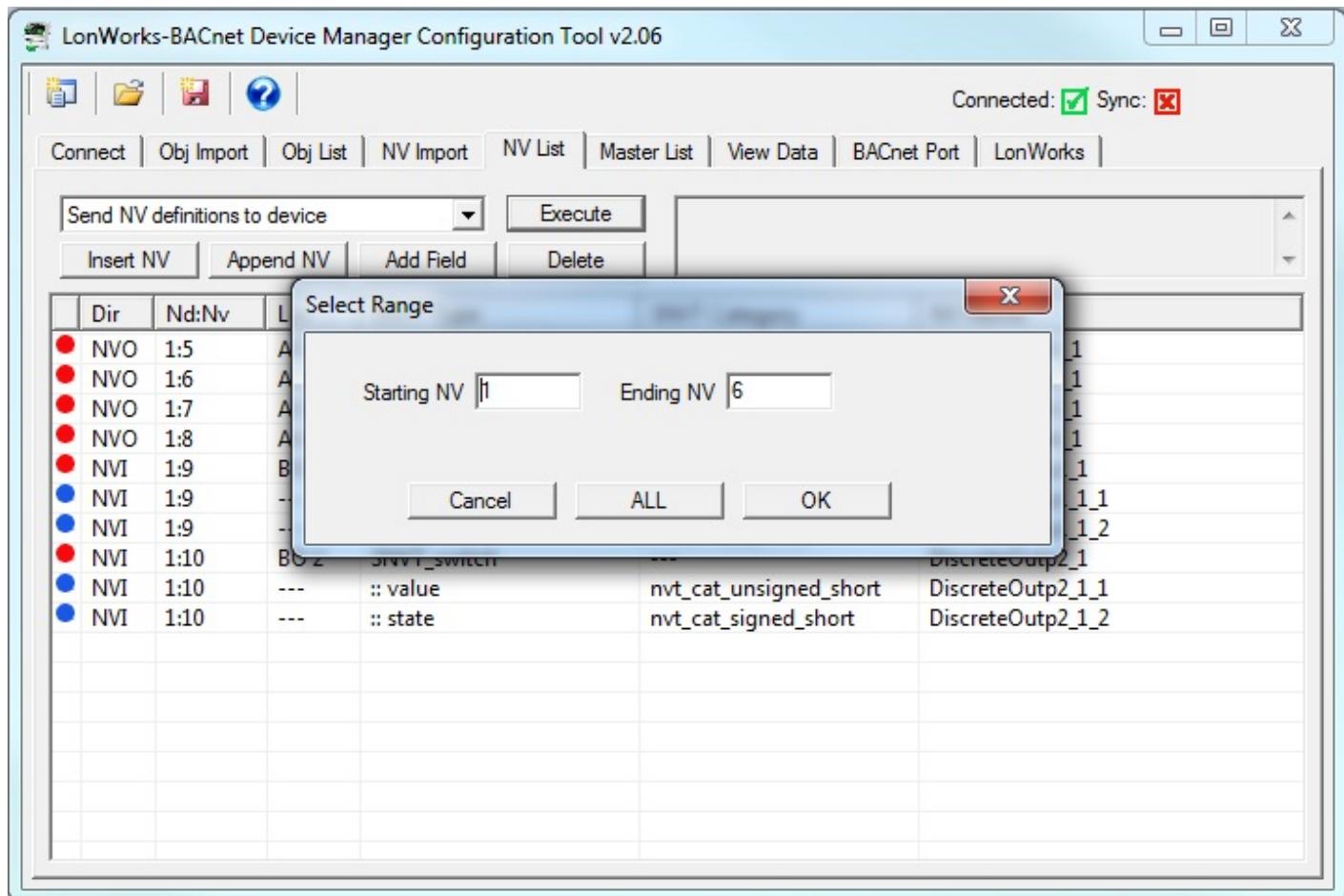
The other thing that must be done is to connect with the LonWorks device identified by node number in the NV List. This is done from the LonWorks page, and the procedure for making this connection is covered in that section of this user guide.



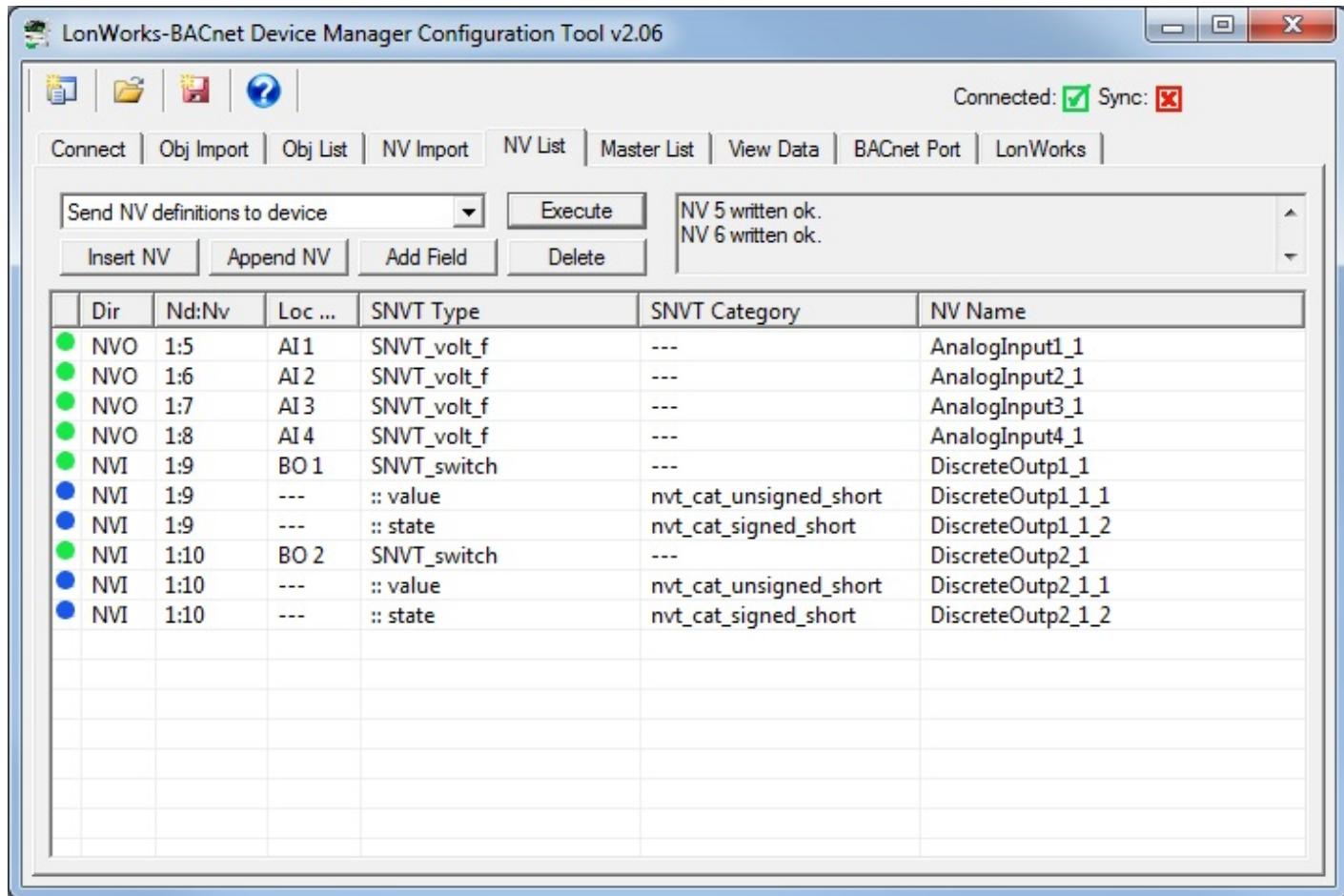
Assuming your node table has been sent to the gateway from the LonWorks page, and the BACnet object counts have been configured from the BACnet Port page, you can proceed to send the updated NV List to the gateway device. Select 'Send NV definitions to device' from the list on the NV List page, and click Execute.



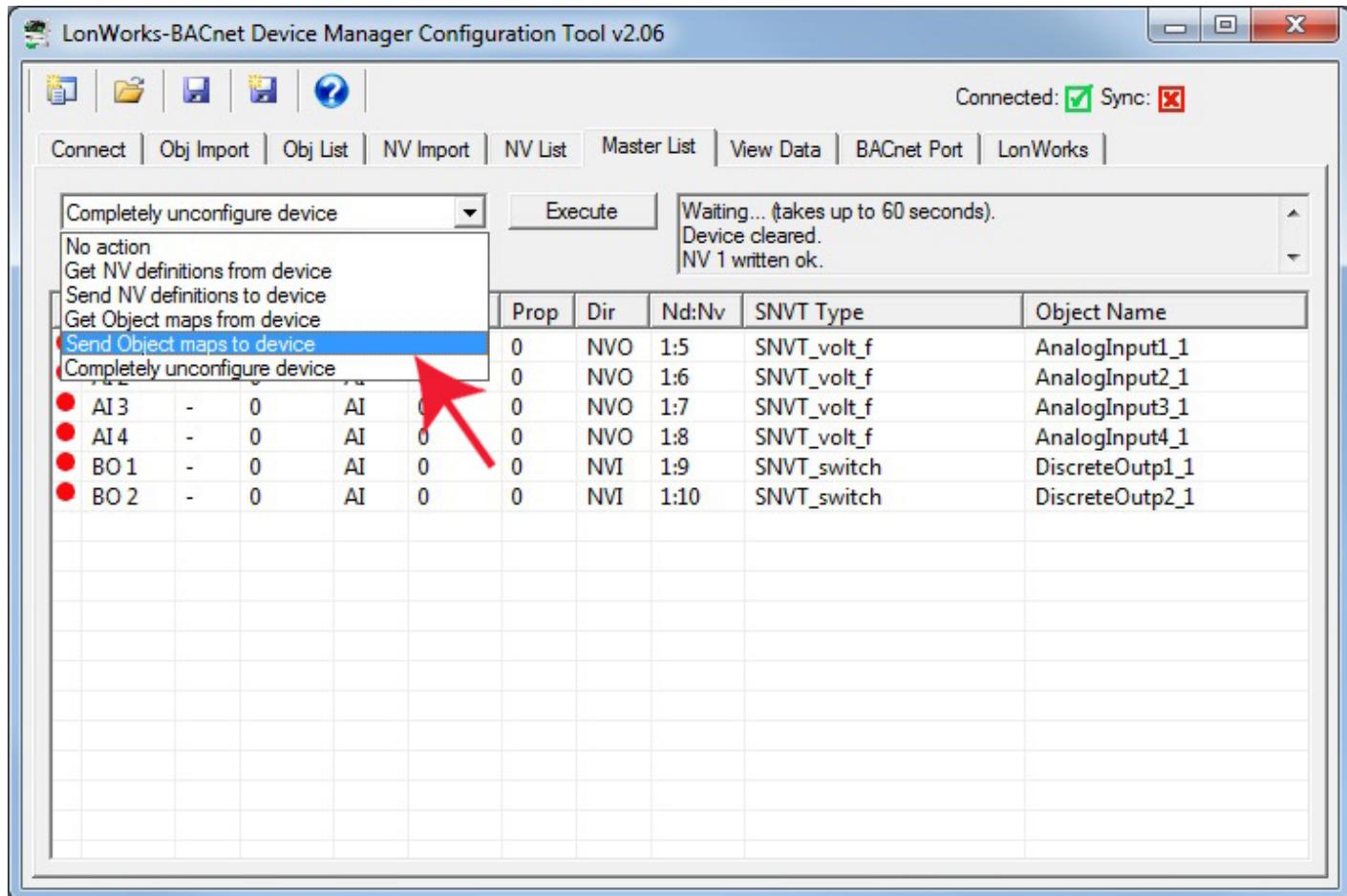
Upon executing 'Send NV definitions to device' you will be given the option of selecting a range. It will default to sending all unconfigured Network Variables, and you can generally just click OK here. If you have previously sent some variables but made changes such that only some remain unconfigured in the device, this dialog will find the range that needs to be sent. If all have been previously sent and you are simply resending, it will default to all Network Variables found in the NV List.



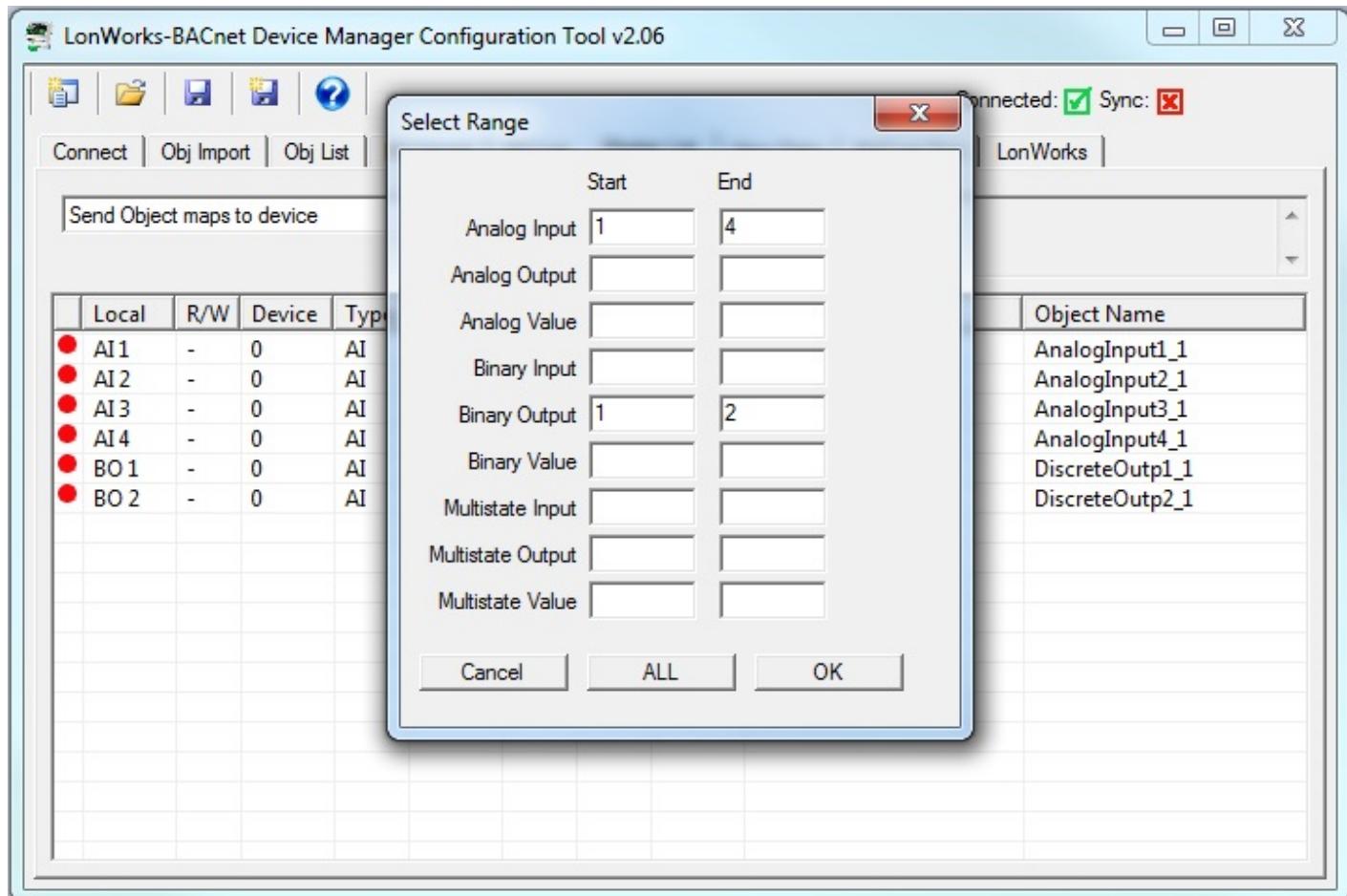
Progress will be indicated as the NV List is being sent.



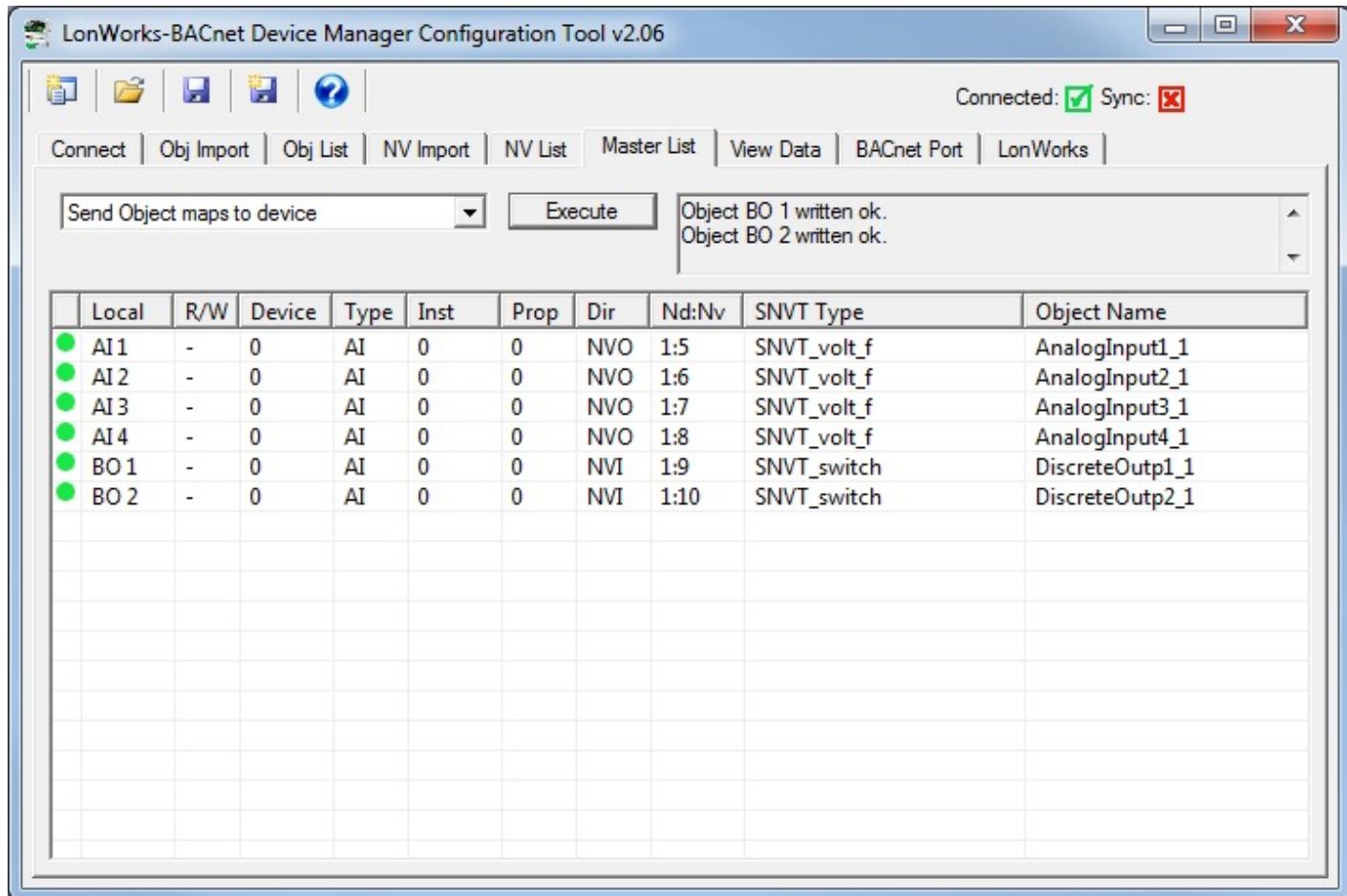
Next, you should proceed to the Master List page and send the object maps. The NV List only defines the list of network variables that will be polled by the gateway. The object maps make the connection between those Network Variables and the BACnet objects you have assigned.



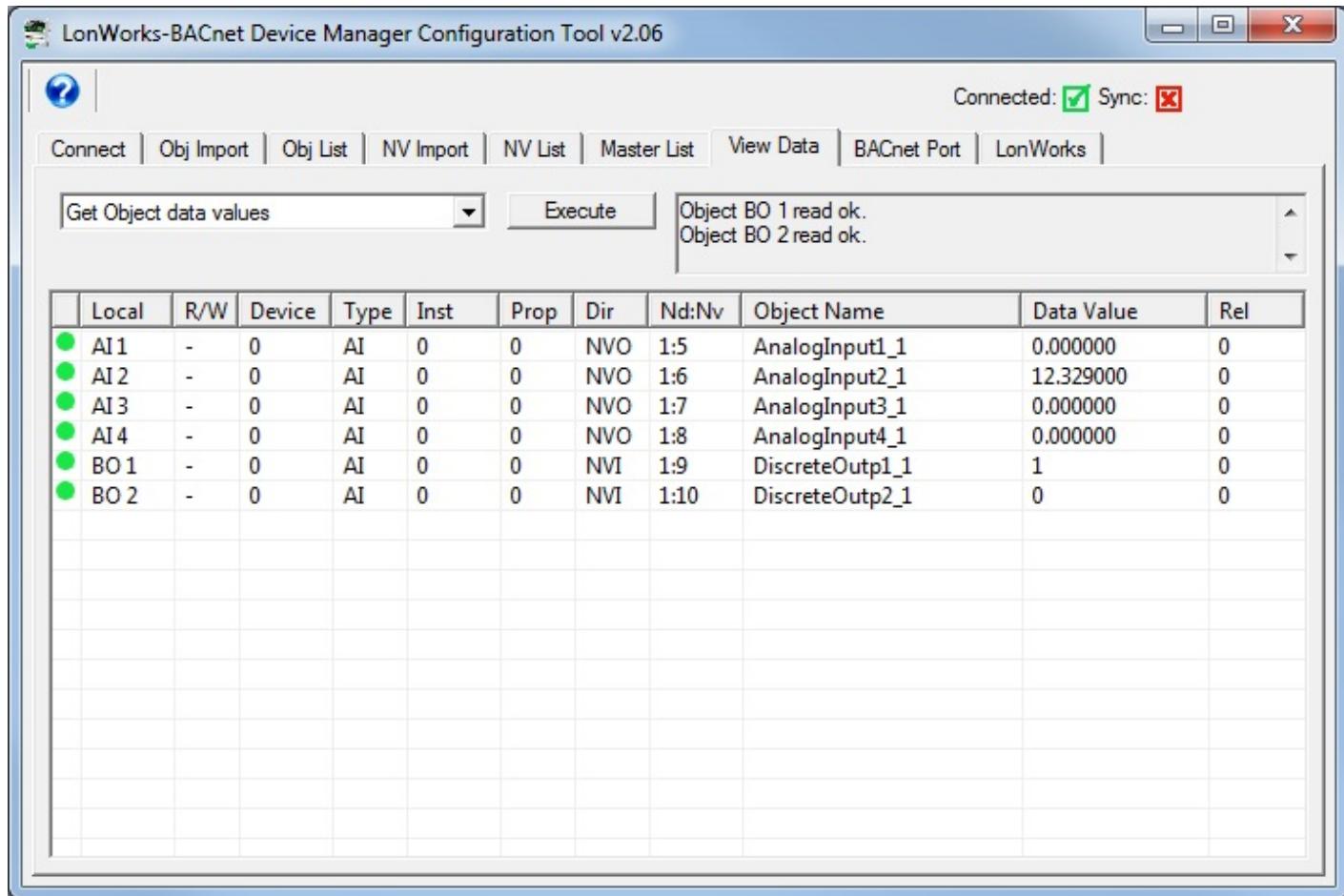
Upon executing 'Send Object maps to device', you will see a dialog pop up with a range of objects to send. There is a range for each of the available BACnet object types supported by the gateway. The range will default to all objects remaining unconfigured, or all objects if resending a list that the tool believes is already completely configured in the device.



Progress will be indicated as objects are sent. Sending object maps takes longer per map than sending NV List entries because there is a lot of information being sent per BACnet object.



Assuming the NV List has been sent, the object maps have been sent from the Master List, and the node is connected from the LonWorks page, you can now view data being obtained from (or sent to) the LonWorks device. Go to the View Data page, select 'Get Object data values' and click Execute. You can also change data written to the LonWorks device from this page (refer to section on View Data page).

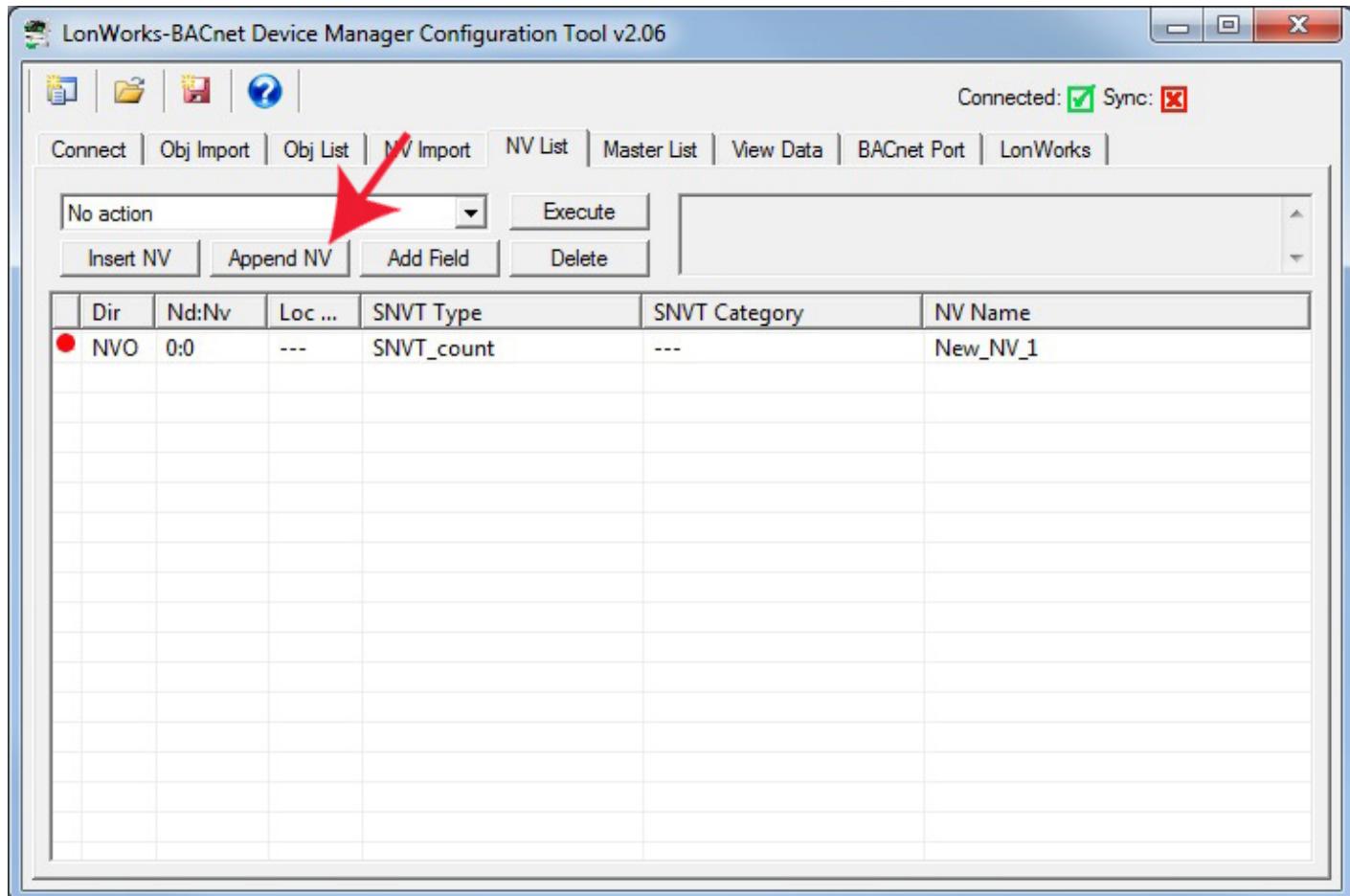


3.4 Build Configuration from Scratch

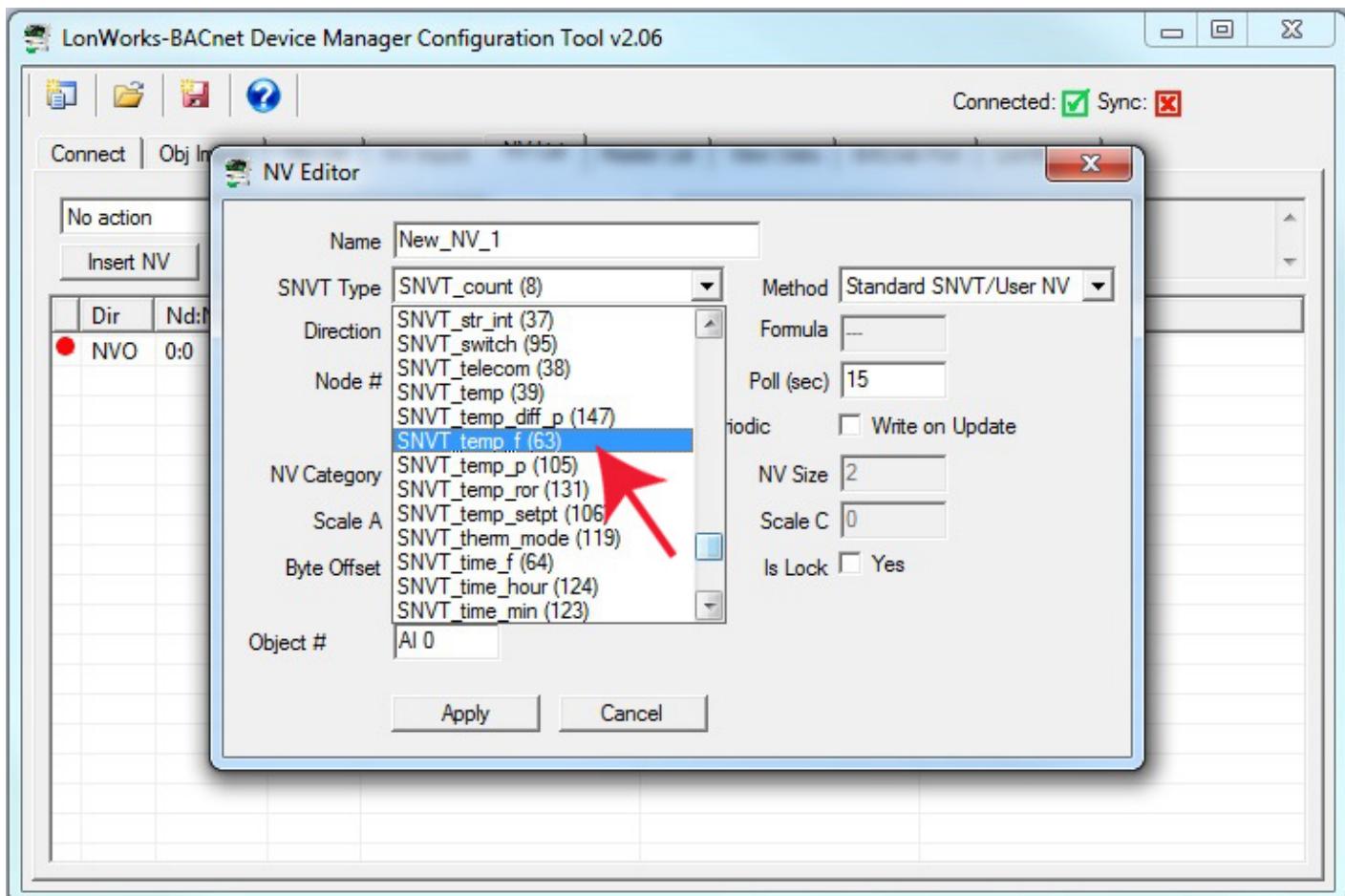
It is not required that you start with either a CSV file or an XIF file. You can use the configuration tool to start with a blank slate and build your configuration from scratch.

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 or Append NV.



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.

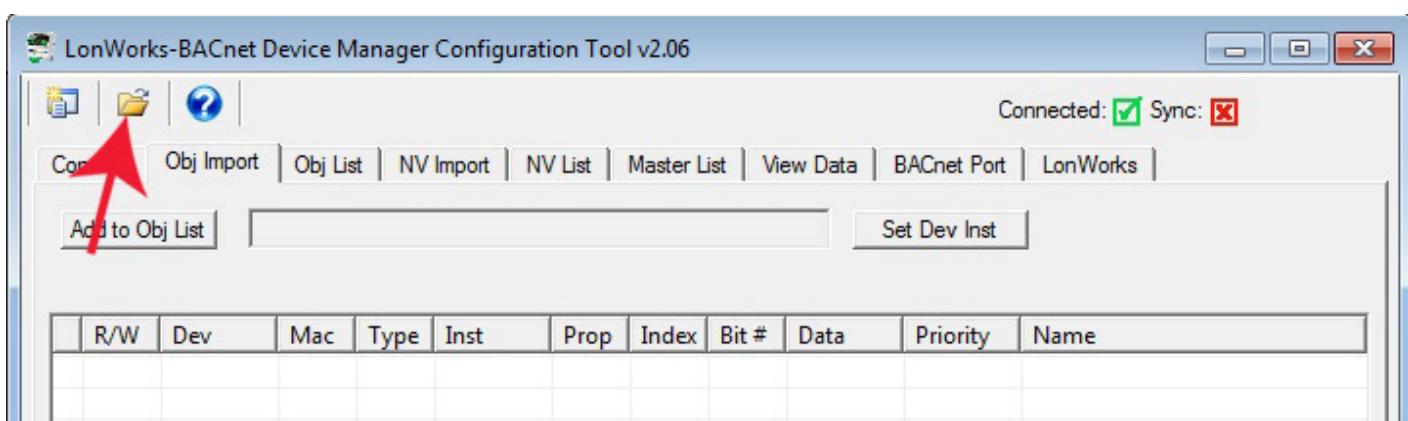


Once you have your list of variables created, you need to assign BACnet objects in the same manner as discussed for building a configuration starting from an XIF file. What you have done by manually entering network variables is replace the XIF import process with a manually imported list. Additional details regarding creation of user defined network variables and structures can be found in the section dedicated to the NV List page.

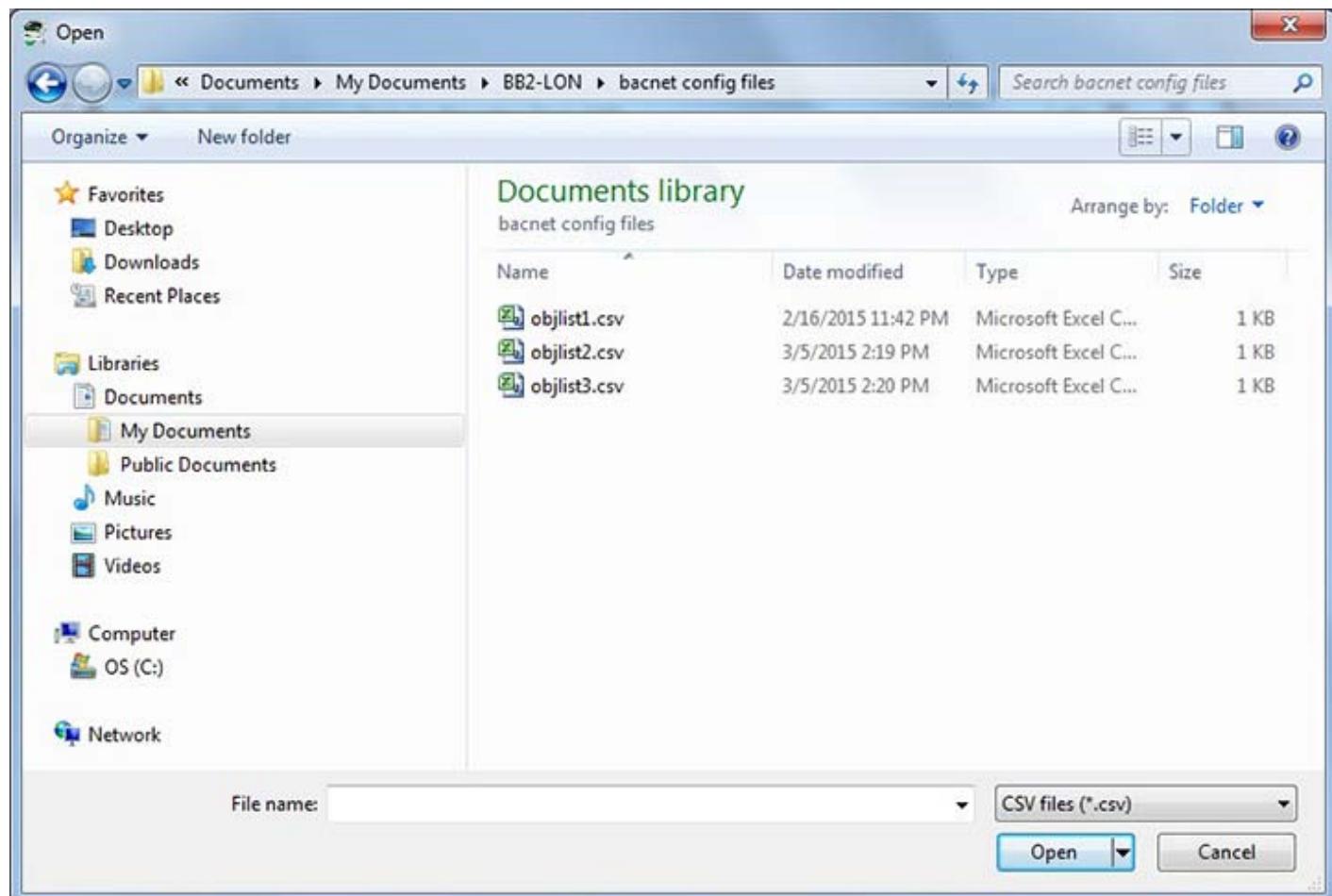
3.5 Build Configuration from CSV List of BACnet Objects

Starting with a BACnet object list for a specific BACnet device is only useful if the Babel Buster gateway will be functioning as a BACnet client. If the Babel Buster will only be a slave (server) on the BACnet network, then the CSV object import is of little use to you.

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.



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
```

R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
R	7171	0	AI	1	85	-1	0	REAL	0	Test AI1
R	7171	0	AI	2	85	-1	0	REAL	0	Test AI2
R	7171	0	AI	3	85	-1	0	REAL	0	Test AI3
W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO1
W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO2
R	7171	0	AV	1	85	-1	0	REAL	0	Test AV1
W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV2
R	7171	0	BI	1	85	-1	0	ENUM	0	Test BI1
R	7171	0	BI	2	85	-1	0	ENUM	0	Test BI2
W+	7171	0	BO	1	85	-1	0	ENUM	0	Test BO1
R	7171	0	BV	1	85	-1	0	ENUM	0	Test BV1
W+	7171	0	BV	2	85	-1	0	ENUM	0	Test BV2
R	7171	0	MI	1	85	-1	0	UINT	0	Test MI1
R	7171	0	MI	2	85	-1	0	UINT	0	Test MI2
W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO1
W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO2
D	7171	0	MV	1	85	-1	0	UINT	0	Test MV1

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.

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.

The screenshot shows the LonWorks-BACnet Device Manager Configuration Tool v2.06. The window title is "LonWorks-BACnet Device Manager Configuration Tool v2.06". The menu bar includes "File", "Edit", "View", "Import", "Tools", and "Help". The toolbar has icons for "Connect", "Obj Import", "Obj List", "NV Import", "NV List", "Master List", "View Data", "BACnet Port", and "LonWorks". The status bar shows "Connected: Sync: ". The main area has tabs for "Obj List" (selected), "NV List", "Master List", "View Data", "BACnet Port", and "LonWorks". A message box says "18 entries added to list." Below it is a "Set Dev Inst" button. A red arrow points to the "Add to Obj List" button. A table lists 18 entries:

R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
R	7171	0	AI	1	85	-1	0	REAL	0	Test AI1
R	7171	0	AI	2	85	-1	0	REAL	0	Test AI2
R	7171	0	AI	3	85	-1	0	REAL	0	Test AI3
W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO1
W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO2
R	7171	0	AV	1	85	-1	0	REAL	0	Test AV1
W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV2
R	7171	0	BI	1	85	-1	0	ENUM	0	Test BI1
R	7171	0	BI	2	85	-1	0	ENUM	0	Test BI2
W+	7171	0	BO	1	85	-1	0	ENUM	0	Test BO1
R	7171	0	BV	1	85	-1	0	ENUM	0	Test BV1
W+	7171	0	BV	2	85	-1	0	ENUM	0	Test BV2
R	7171	0	MI	1	85	-1	0	UINT	0	Test MI1
R	7171	0	MI	2	85	-1	0	UINT	0	Test MI2
W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO1
W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO2
D	7171	0	MV	1	85	-1	0	UINT	0	Test MV1

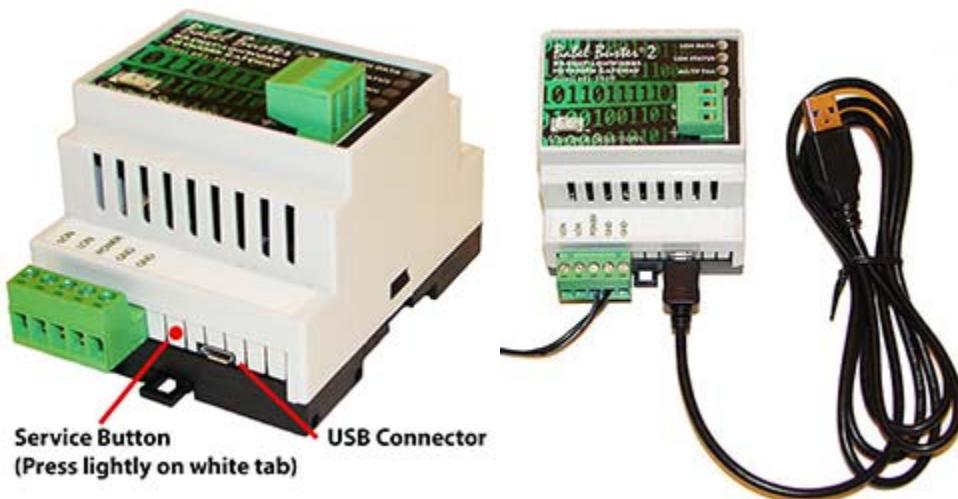
Once you have imported your object list(s), go to the Obj List page. Here is where you begin the auto-build of the rest of your configuration. Refer to the Obj List page section of this user guide to proceed from here.



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-NB and BB2-7020-NB. The only significant difference is that the IP settings will show up instead of MS/TP settings on the BACnet Port page when BB2-7020-NB is selected as Device Model.

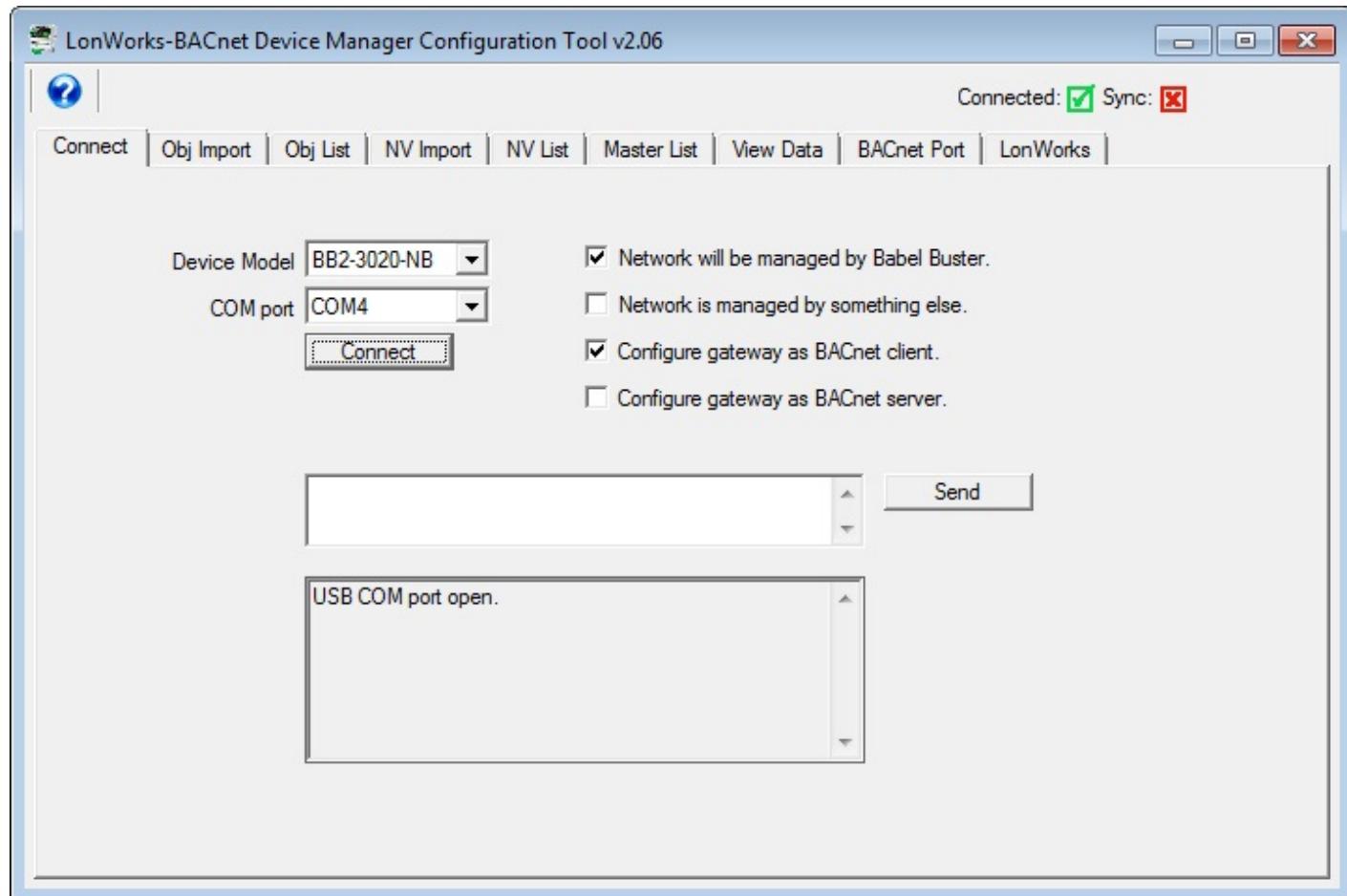
If the BB2-3020-NB or BB2-7020-NB will be managing the devices on the LonWorks side, then leave the default selection of "Network will be managed by Babel Buster". If the LonWorks devices are on a managed network that has been commissioned and bound by a tool such as Echelon's LonMaker, then select "Network is managed by something else".

IMPORTANT: If the LonWorks network has been commissioned and bound by a network management tool (e.g. LonMaker) and the LonWorks network is expected to remain functional, then you MUST select "Network is managed by something else". If you allow Babel Buster to attempt to manage a network that is already managed by something else, YOU WILL BREAK THE NETWORK!

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.

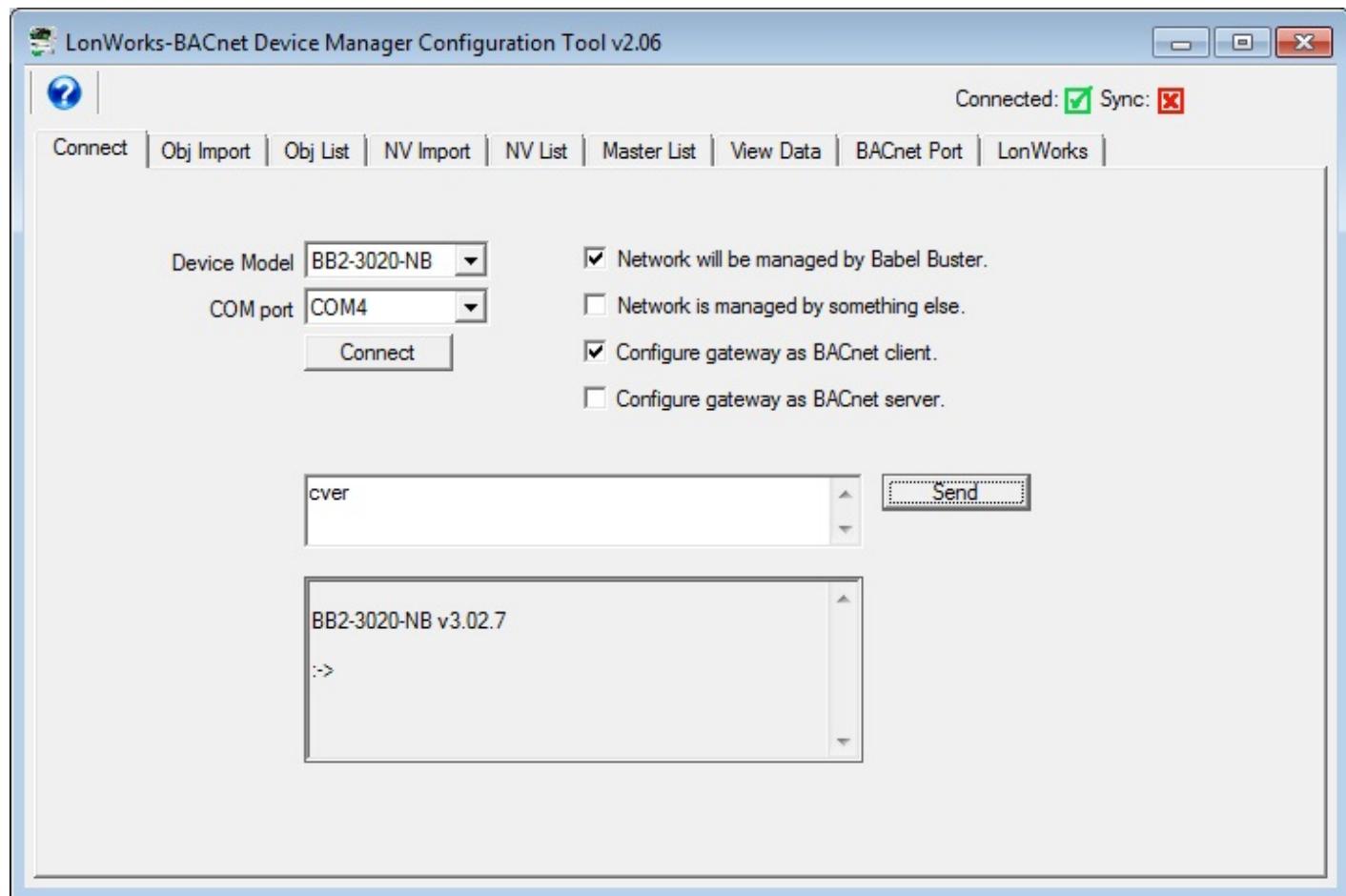
IMPORTANT: If you have connected USB to the gateway, and then power cycle the gateway, the USB connection will be lost. You will need to unplug the USB cable from the gateway and reconnect, then reconnect the configuration tool. It may also sometimes be necessary to close and restart the configuration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



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.



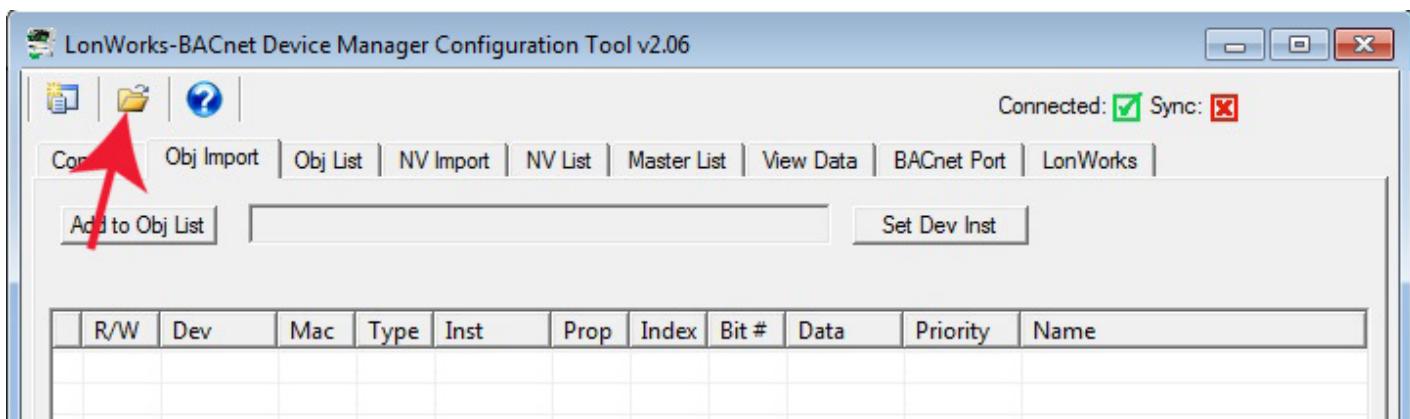


5 Tool 'Obj Import' Page

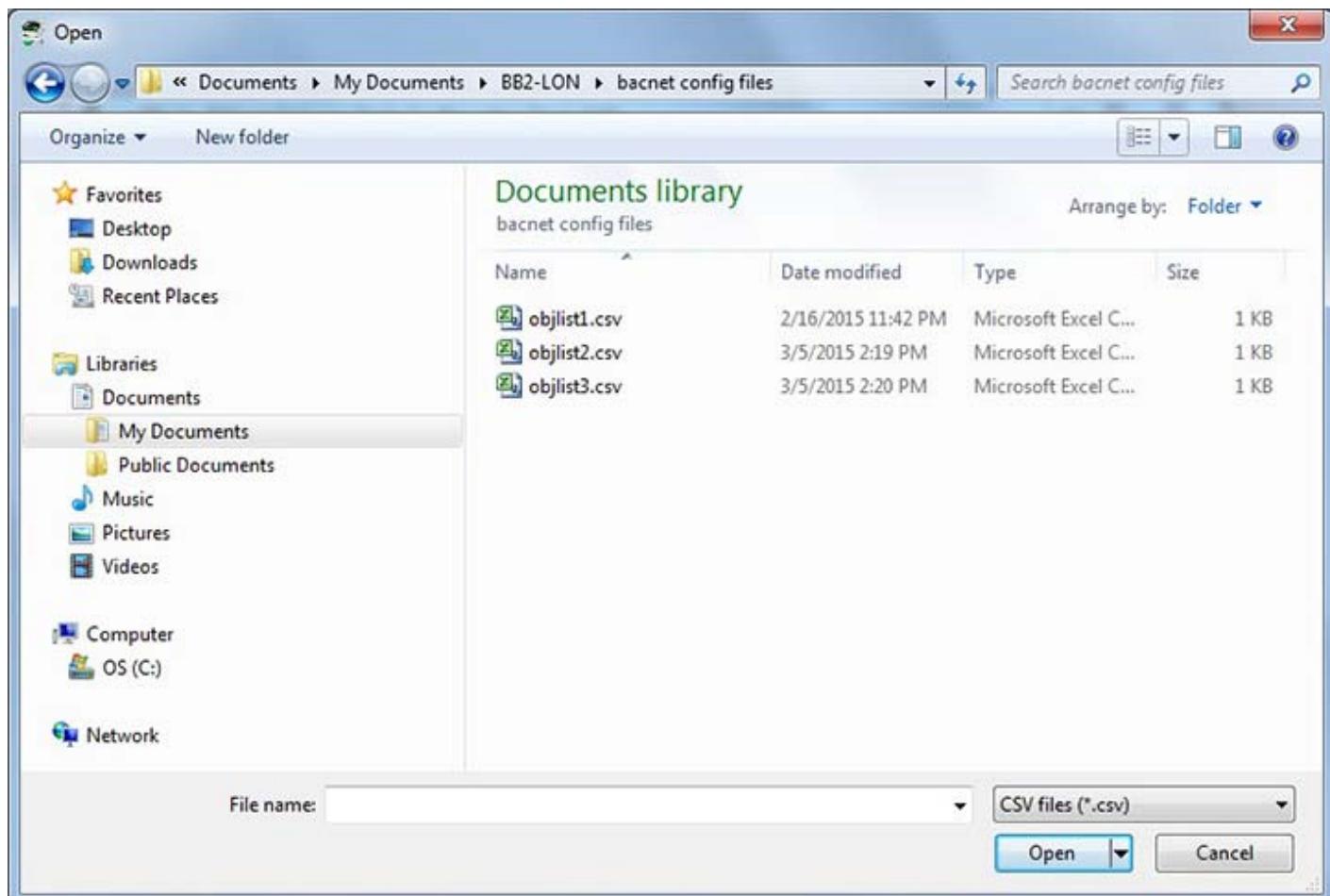
5.1 Importing a CSV Object List

The purpose of the object import capability is to allow you to use a standard spreadsheet program to create object mappings, and probably more importantly allow you to edit those mappings easily. The object mappings are used when the gateway will function as a BACnet client (master). If your gateway will function as a BACnet server (slave) only, then you do not have any need to import objects here. The "objects" imported here are a list of objects in other BACnet devices that the Babel Buster gateway, function as BACnet client, will poll, meaning read or write according to the rules you define. Once the object mappings are imported, you will still need to assign local objects on the "Obj List" page. Refer to Appendix D for a reference guide on the format of this CSV file.

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.



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
```

The screenshot shows the LonWorks-BACnet Device Manager Configuration Tool v2.06 window. The title bar reads "LonWorks-BACnet Device Manager Configuration Tool v2.06". The menu bar includes "File", "Edit", "Import", "Object List", "NV List", "Master List", "View Data", "BACnet Port", and "LonWorks". The toolbar has icons for "New", "Open", "Save", and "Help". The status bar shows "Connected: Sync: ". The main area has tabs: "Connect", "Obj Import" (selected), "Obj List", "NV Import", "NV List", "Master List", "View Data", "BACnet Port", and "LonWorks". A message box says "Add to Obj List" and "18 objects parsed from objlist3.csv.". A "Set Dev Inst" button is also present. Below is a table with columns: R/W, Dev, Mac, Type, Inst, Prop, Index, Bit #, Data, Priority, and Name. The data rows are as follows:

R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
R	7171	0	AI	1	85	-1	0	REAL	0	Test AI1
R	7171	0	AI	2	85	-1	0	REAL	0	Test AI2
R	7171	0	AI	3	85	-1	0	REAL	0	Test AI3
W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO1
W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO2
R	7171	0	AV	1	85	-1	0	REAL	0	Test AV1
W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV2
R	7171	0	BI	1	85	-1	0	ENUM	0	Test BI1
R	7171	0	BI	2	85	-1	0	ENUM	0	Test BI2
W+	7171	0	BO	1	85	-1	0	ENUM	0	Test BO1
R	7171	0	BV	1	85	-1	0	ENUM	0	Test BV1
W+	7171	0	BV	2	85	-1	0	ENUM	0	Test BV2
R	7171	0	MI	1	85	-1	0	UINT	0	Test MI1
R	7171	0	MI	2	85	-1	0	UINT	0	Test MI2
W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO1
W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO2
D	7171	0	MV	1	85	-1	0	UINT	0	Test MV1

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.

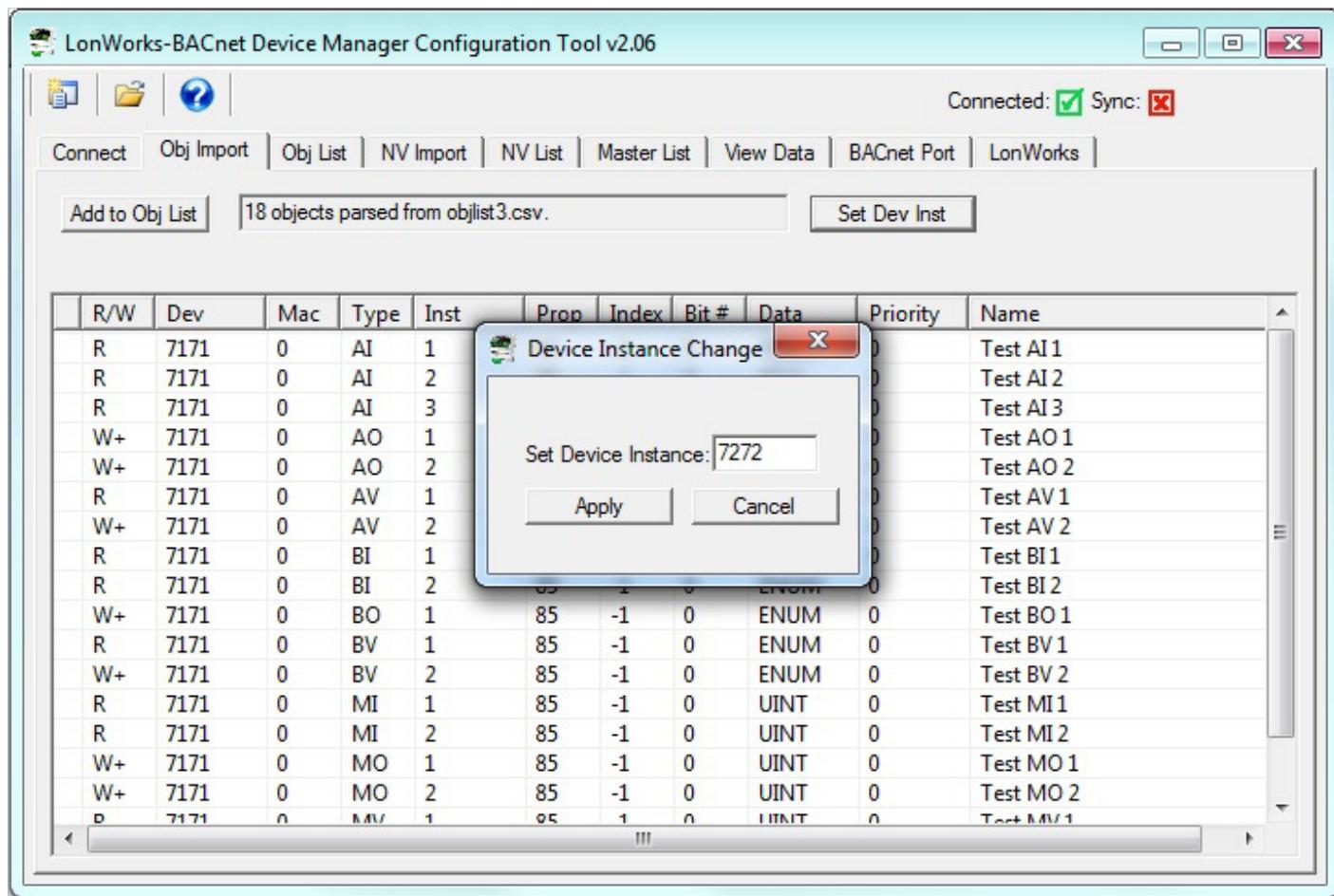
To select all objects, click the icon column header as illustrated by the red arrow below. This will cause the blue dot to appear on each line, which indicates that this object has been selected to be included in the configuration. To select only certain objects, click the icon area of only those rows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

The screenshot shows the LonWorks-BACnet Device Manager Configuration Tool v2.06. The interface includes a menu bar with File, Edit, View, Tools, Help, and a toolbar with icons for New, Open, Save, Print, and Help. The main window has tabs for Connect, Obj Import, Obj List, NV Import, NV List, Master List, View Data, BACnet Port, and LonWorks. A status bar at the top right shows 'Connected: Sync: '. Below the tabs, a message box says '18 entries added to list.' with a 'Set Dev Inst' button. A red arrow points to the 'Add to Obj List' button. The central part of the window is a table with columns: R/W, Dev, Mac, Type, Inst, Prop, Index, Bit #, Data, Priority, and Name. The table lists 18 entries, mostly starting with 'R' or 'W+' followed by device ID 7171 and various object types like AI, AO, AV, BI, BO, BV, MI, MO, and MV.

R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
R	7171	0	AI	1	85	-1	0	REAL	0	Test AI1
R	7171	0	AI	2	85	-1	0	REAL	0	Test AI2
R	7171	0	AI	3	85	-1	0	REAL	0	Test AI3
W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO1
W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO2
R	7171	0	AV	1	85	-1	0	REAL	0	Test AV1
W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV2
R	7171	0	BI	1	85	-1	0	ENUM	0	Test BI1
R	7171	0	BI	2	85	-1	0	ENUM	0	Test BI2
W+	7171	0	BO	1	85	-1	0	ENUM	0	Test BO1
R	7171	0	BV	1	85	-1	0	ENUM	0	Test BV1
W+	7171	0	BV	2	85	-1	0	ENUM	0	Test BV2
R	7171	0	MI	1	85	-1	0	UINT	0	Test MI1
R	7171	0	MI	2	85	-1	0	UINT	0	Test MI2
W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO1
W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO2
D	7171	0	MV	1	85	-1	0	UINT	0	Test MV1

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.



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 if you are starting from a CSV list of BACnet client mappings. Start by selecting "Auto-allocate Network Variable Maps". Then click the Execute button.

Note: If you are starting your configuration from a BACnet client map list, the network variables automatically allocated might not match the LonWorks device you will be communicating with. You will need to manually edit the network variables as applicable. The more common and possibly more useful approach is to start on the NV Import page. The BACnet object list import approach would be used if you have a specific list of BACnet objects that you require matching, and is only useful if the Babel Buster gateway will be operating as a BACnet client. If the gateway will be a BACnet server (slave), then matching network variables is more important and you should start on the NV Import page instead.

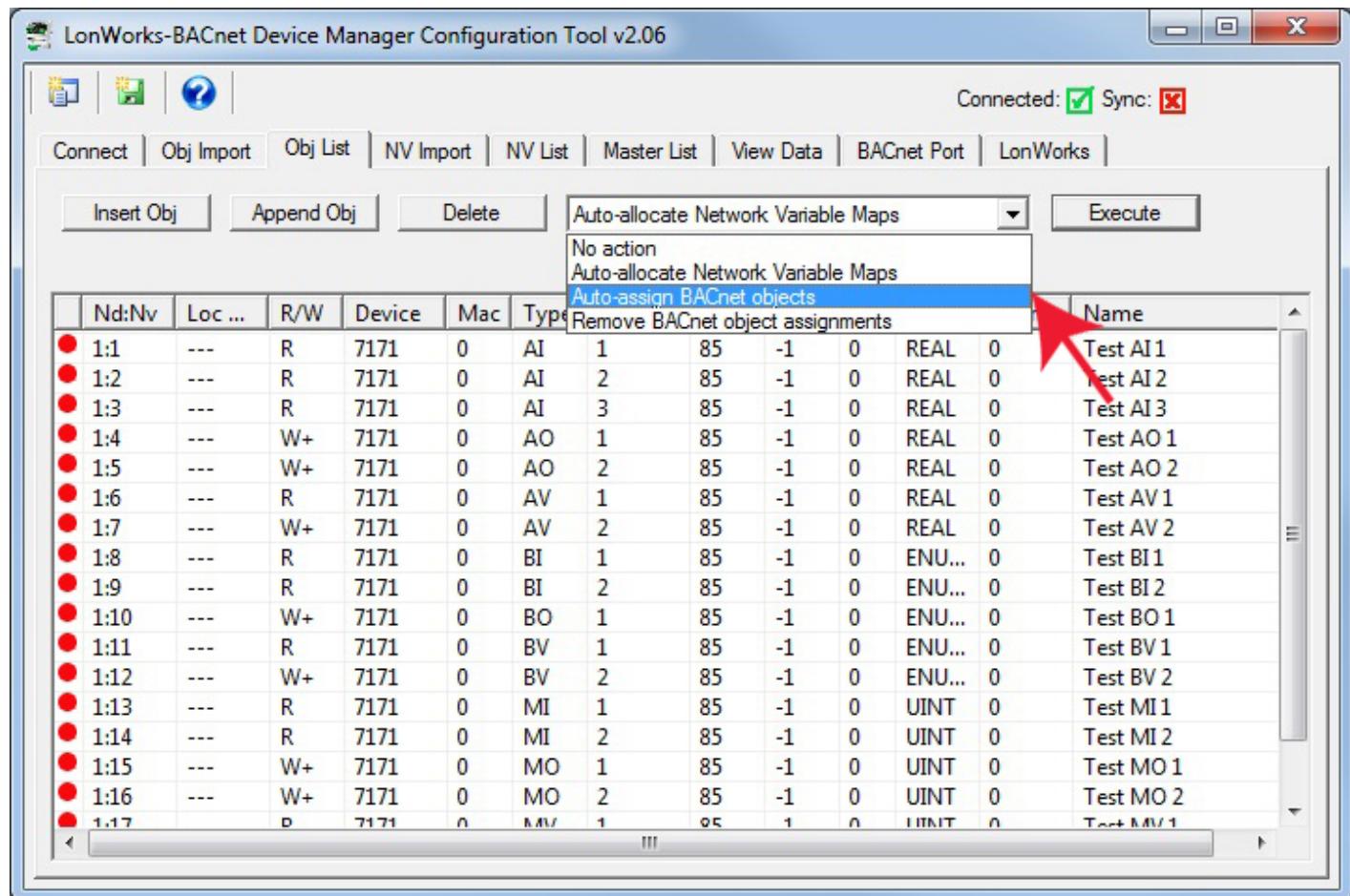
The screenshot shows the 'Obj List' tab selected in the tool's navigation bar. Below the tabs, there are buttons for 'Insert Obj', 'Append Obj', 'Delete', and a dropdown menu currently set to 'No action'. A red arrow points to the 'Execute' button, which has a dropdown menu open with the option 'Auto-allocate Network Variable Maps' highlighted. The main area of the window displays a table of BACnet objects with columns for Nd:Nv, Loc ..., R/W, Device, Mac, Type, and Name. The 'Name' column contains entries like 'Test AI1', 'Test AI2', etc.

Nd:Nv	Loc ...	R/W	Device	Mac	Type	Name
---	---	R	7171	0	AI	Test AI1
---	---	R	7171	0	AI	Test AI2
---	---	R	7171	0	AI	Test AI3
---	---	W+	7171	0	AO	Test AO1
---	---	W+	7171	0	AO	Test AO2
---	---	R	7171	0	AV	Test AV1
---	---	W+	7171	0	AV	Test AV2
---	---	R	7171	0	BI	Test BI1
---	---	R	7171	0	BI	Test BI2
---	---	W+	7171	0	BO	Test BO1
---	---	R	7171	0	BV	Test BV1
---	---	W+	7171	0	BV	Test BV2
---	---	R	7171	0	MI	Test MI1
---	---	R	7171	0	MI	Test MI2
---	---	W+	7171	0	MO	Test MO1
---	---	W+	7171	0	MO	Test MO2
---	---	D	7171	0	MI	Test MI1

Upon executing the Auto-allocate, the "Nd:Nv" column will be populated. The first number (Nd) is node

number, and identifies a node in the node list on the LonWorks page. The second number (Nv) is the network variable index expected to be found in that node. The "R/W" designation indicates whether the BACnet object will be read from or written to in the remote BACnet device being mapped here.

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



Upon executing Auto-assign BACnet objects, the object numbers allocated will appear in the Loc... (Local Object) column. The object numbers in this column are the local BACnet objects found in the gateway itself. When operating as a BACnet client, the gateway will read BACnet objects from other devices and place the data received in the local object shown.

Nd:Nv	Loc ...	R/W	Device	Mac	Type	Inst	Prop	Index	Bit	Data	Priority	Name
1:1	AI1	R	7171	0	AI	1	85	-1	0	REAL	0	Test AI1
1:2	AI2	R	7171	0	AI	2	85	-1	0	REAL	0	Test AI2
1:3	AI3	R	7171	0	AI	3	85	-1	0	REAL	0	Test AI3
1:4	AO1	W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO1
1:5	AO2	W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO2
1:6	AI4	R	7171	0	AV	1	85	-1	0	REAL	0	Test AV1
1:7	AO3	W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV2
1:8	AI5	R	7171	0	BI	1	85	-1	0	ENU...	0	Test BI1
1:9	AI6	R	7171	0	BI	2	85	-1	0	ENU...	0	Test BI2
1:10	AO4	W+	7171	0	BO	1	85	-1	0	ENU...	0	Test BO1
1:11	AI7	R	7171	0	BV	1	85	-1	0	ENU...	0	Test BV1
1:12	AO5	W+	7171	0	BV	2	85	-1	0	ENU...	0	Test BV2
1:13	AI8	R	7171	0	MI	1	85	-1	0	UINT	0	Test MI1
1:14	AI9	R	7171	0	MI	2	85	-1	0	UINT	0	Test MI2
1:15	AO6	W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO1
1:16	AO7	W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO2
1:17	AI10	D	7171	0	MV	1	85	-1	0	UINT	0	Test MV1

With the exception of making sure the network variables actually match your LonWorks device, 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. If the network variables match your LonWorks device, then 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.

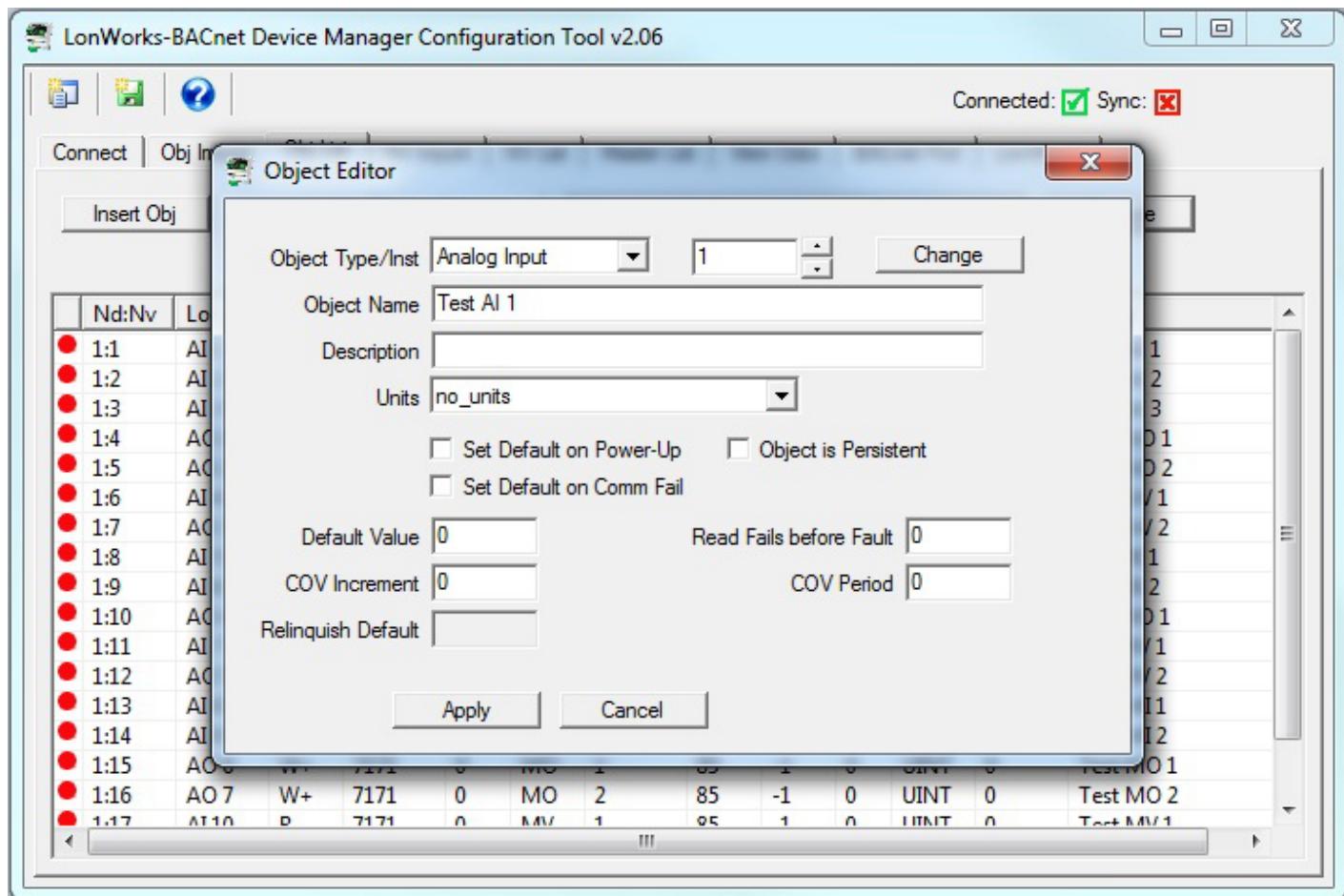
6.2 Editing the Object List

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 BACnet client).

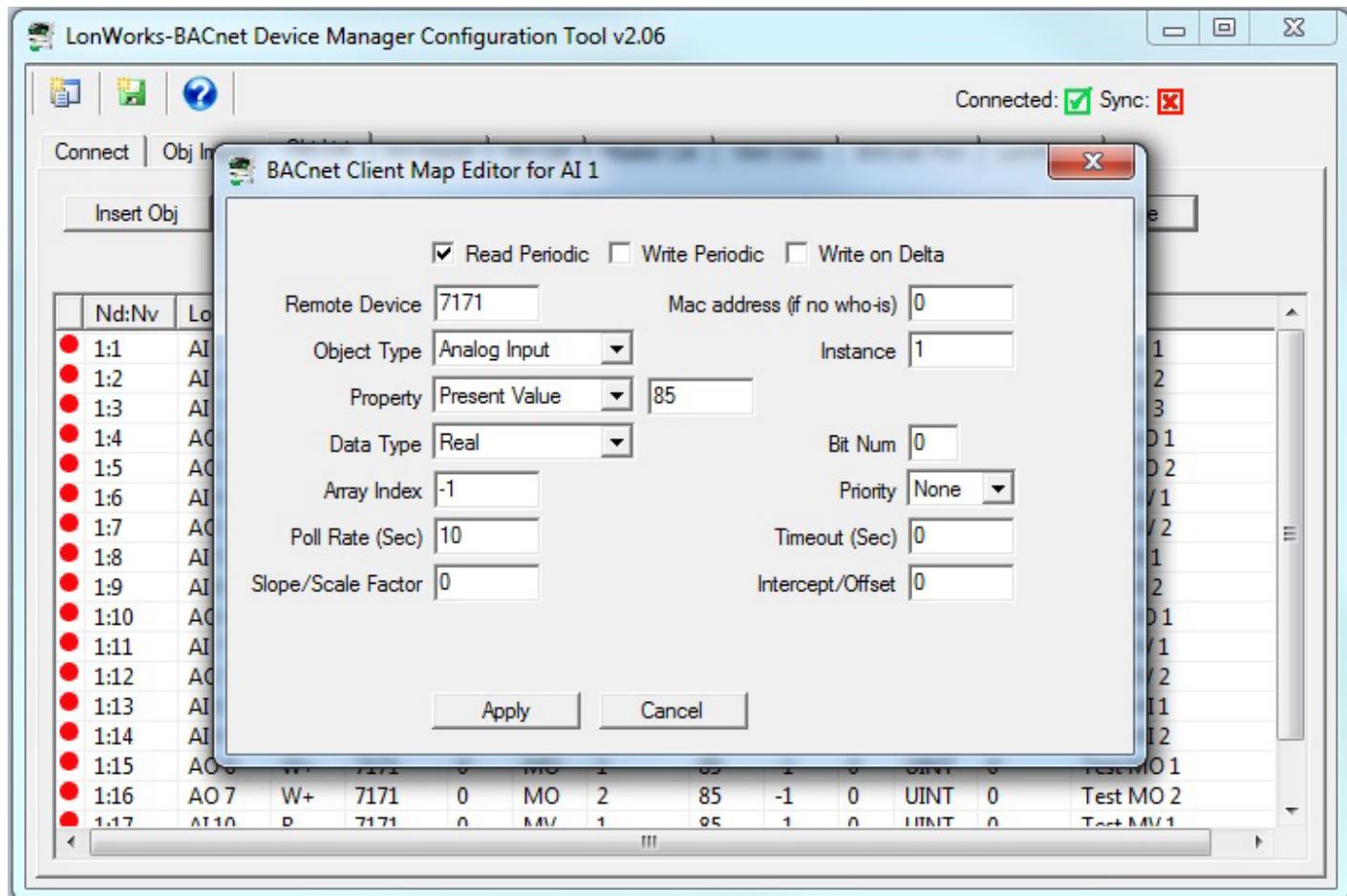
The screenshot shows the LonWorks-BACnet Device Manager Configuration Tool interface. The title bar reads "LonWorks-BACnet Device Manager Configuration Tool v2.06". The menu bar includes "Connect", "Obj Import", "Obj List" (which is selected), "NV Import", "NV List", "Master List", "View Data", "BACnet Port", and "LonWorks". Below the menu is a toolbar with icons for "Insert Obj", "Append Obj", "Delete", and "Auto-assign BACnet objects" (with a dropdown arrow) and an "Execute" button. The main area is a table titled "Obj List" with the following columns: Nd:Nv, Loc ..., R/W, Device, Mac, Type, Inst, Prop, Index, Bit, Data, Priority, and Name. The table contains 17 rows of data, each representing a BACnet object. The first row, "1:1 AI1 R", is highlighted with a blue background. A red arrow points to the "Nd:Nv" and "Loc ..." columns of this row.

Nd:Nv	Loc ...	R/W	Device	Mac	Type	Inst	Prop	Index	Bit	Data	Priority	Name
1:1	AI1	R	7171	0	AI	1	85	-1	0	REAL	0	Test AI1
1:2	AI2	R	7171	0	AI	2	85	-1	0	REAL	0	Test AI2
1:3	AI3	R	7171	0	AI	3	85	-1	0	REAL	0	Test AI3
1:4	AO1	W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO1
1:5	AO2	W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO2
1:6	AI4	R	7171	0	AV	1	85	-1	0	REAL	0	Test AV1
1:7	AO3	W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV2
1:8	AI5	R	7171	0	BI	1	85	-1	0	ENU...	0	Test BI1
1:9	AI6	R	7171	0	BI	2	85	-1	0	ENU...	0	Test BI2
1:10	AO4	W+	7171	0	BO	1	85	-1	0	ENU...	0	Test BO1
1:11	AI7	R	7171	0	BV	1	85	-1	0	ENU...	0	Test BV1
1:12	AO5	W+	7171	0	BV	2	85	-1	0	ENU...	0	Test BV2
1:13	AI8	R	7171	0	MI	1	85	-1	0	UINT	0	Test MI1
1:14	AI9	R	7171	0	MI	2	85	-1	0	UINT	0	Test MI2
1:15	AO6	W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO1
1:16	AO7	W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO2
1:17	AI10	R	7171	0	MV	1	85	-1	0	UINT	0	Test MV1

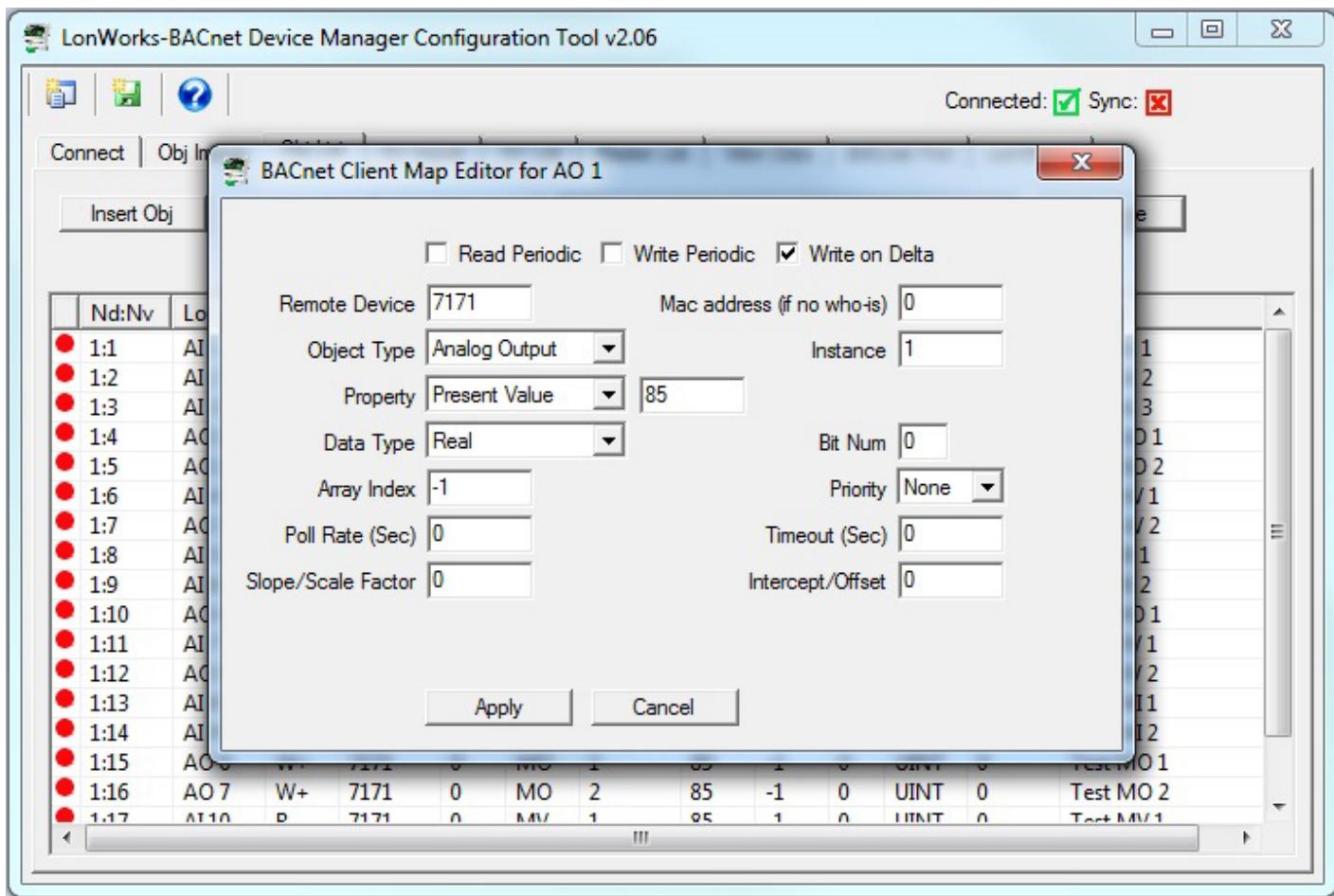
Upon double clicking the first couple of columns in a line on the Obj List, the dialog shown below will pop up for editing the local BACnet object.



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.

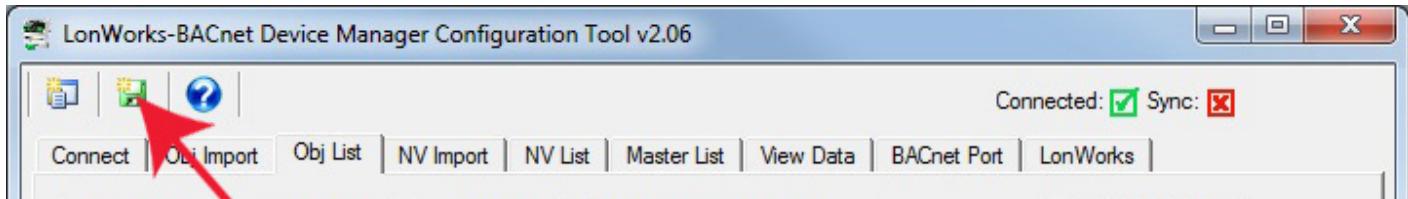


The second example (below) is for writing to an Analog Output in another BACnet device.



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.

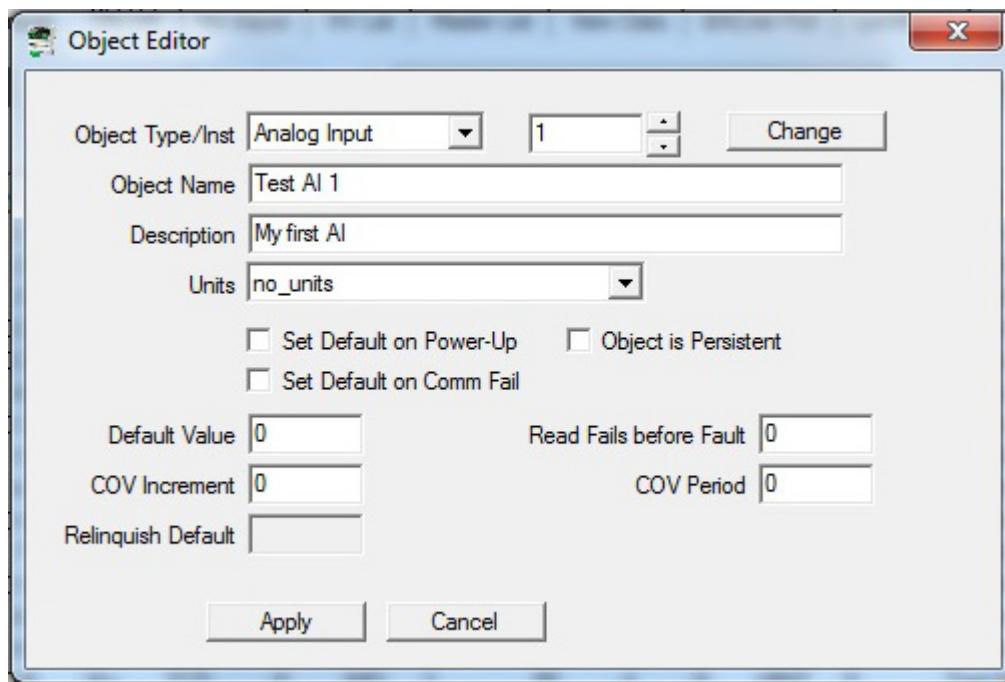
Important: The CSV file created here only saves the BACnet client mappings. To save an entire configuration including network variable definitions, go to the Master List page and click the new file icon there. The file icons are different colors on different pages to call attention to the fact that each different file icon saves a different type of file.

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.



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. To change an object 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 message will indicate 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.

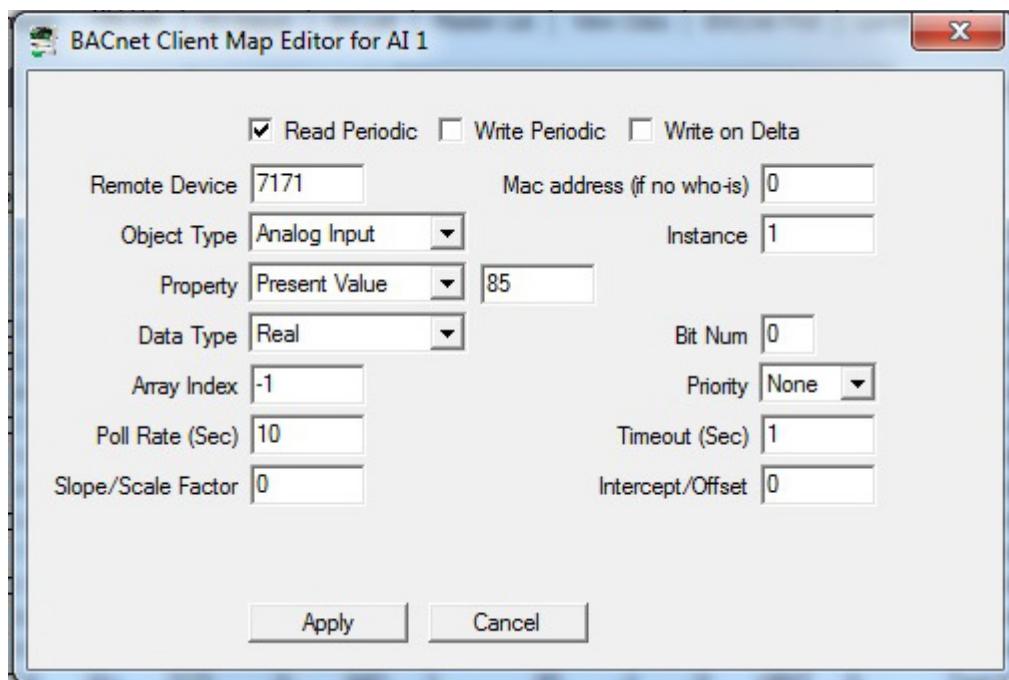
Object Type/Inst	Select BACnet object type from the list, and enter or scroll to the desired object instance. The object number will stop scrolling at the limit of the configuration tool, but this instance could still exceed the object limit of the device - you will need to check device instance limits 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.



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.



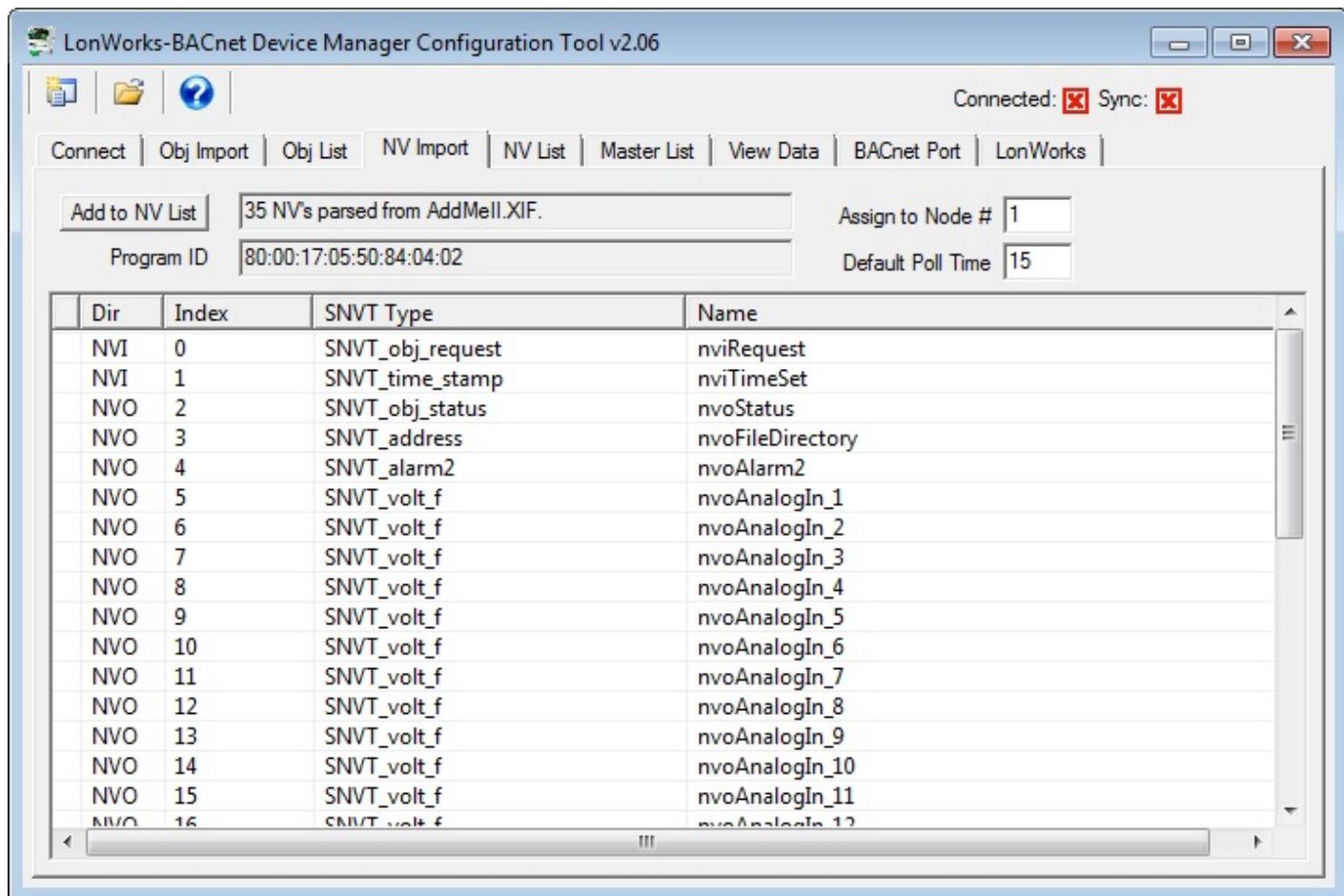
7 Tool 'NV Import' Page

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 LonWorks device accessible as a BACnet device on a BACnet network.

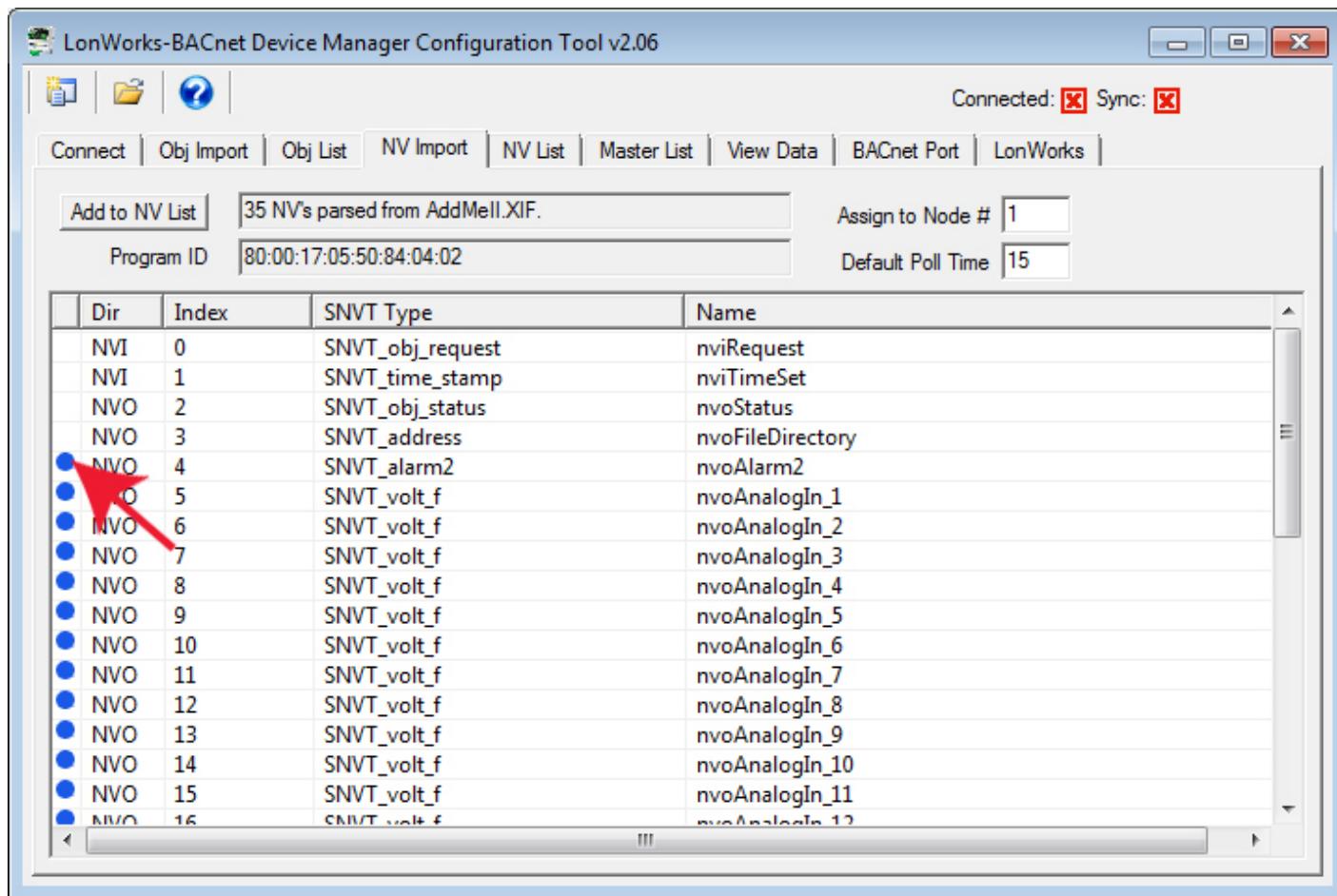
To begin building your configuration from an XIF file, go to the NV Import page. Click on the file icon to open an XIF file. Once the XIF is imported, the list of network variables will be displayed on the NV Import page.

Note: The XIF file is something you would obtain from the manufacturer of the LonWorks device. You can also use Echelon's NodeUtil tool to retrieve the XIF file from the device itself if you have a LonWorks network interface available on your PC. The third option is to use the Babel Buster gateway in conjunction with the configuration tool to retrieve the XIF information from the device. Although slightly more cumbersome, using the gateway to discover network variables in the device is an option and you would start this process on the LonWorks page of the configuration tool.



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.



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 will never actually contain 1000 network variables. The BB2-3020-NB or BB2-7020-NB has a network variable mapping capacity of 300 network 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 300 LonWorks network variables or 300 BACnet objects (combined total of all object types).



8 Tool 'NV List' Page

8.1 Configuration from XIF File

Regardless of whether you imported an XIF file starting on the NV Import page, or imported the XIF information from the device starting on the LonWorks page, you will end up with a list of network variables on the NV List page, and need to continue the configuration process here.

The NV List is the definition of the list of network variables in other LonWorks devices that will be polled by the gateway. 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.

 A screenshot of the LonWorks-BACnet Device Manager Configuration Tool window. The title bar reads "LonWorks-BACnet Device Manager Configuration Tool v2.06". The menu bar includes "File", "Edit", "View", "Tools", "Help", and "About". The toolbar contains icons for "Connect", "Import", "List", "Import", "NV List", "Master List", "View Data", "BACnet Port", and "LonWorks". The status bar shows "Connected: Sync: ". The main interface has tabs: "Connect", "Obj Import", "Obj List", "NV Import", "NV List" (which is selected), "Master List", "View Data", "BACnet Port", and "LonWorks". Below the tabs is a toolbar with buttons for "No action", "Execute", "Insert NV", "Append NV", "Add Field", and "Delete". The main area is a table titled "NV List" with columns: Dir, Nd:Nv, Loc ..., SNVT Type, SNVT Category, and NV Name. The table lists 14 rows of network variable definitions, each with a red circular icon in the first column.

Dir	Nd:Nv	Loc ...	SNVT Type	SNVT Category	NV Name
● NVO	1:4	---	SNVT_alarm2	---	nvoAlarm2
● NVO	1:4	---	:: alarm_type	nvt_cat_enum	nvoAlarm2_1
● NVO	1:4	---	:: priority_level	nvt_cat_enum	nvoAlarm2_2
● NVO	1:4	---	:: alarm_time	nvt_cat_signed_quad	nvoAlarm2_3
● NVO	1:4	---	:: milliseconds	nvt_cat_signed_long	nvoAlarm2_4
● NVO	1:4	---	:: sequence_number	nvt_cat_unsigned_short	nvoAlarm2_5
● NVO	1:4	---	:: description	nvt_cat_unsigned_char	nvoAlarm2_6
● NVO	1:5	---	SNVT_volt_f	---	nvoAnalogIn_1
● NVO	1:6	---	SNVT_volt_f	---	nvoAnalogIn_2
● NVO	1:7	---	SNVT_volt_f	---	nvoAnalogIn_3
● NVO	1:8	---	SNVT_volt_f	---	nvoAnalogIn_4
● NVO	1:9	---	SNVT_volt_f	---	nvoAnalogIn_5
● NVO	1:10	---	SNVT_volt_f	---	nvoAnalogIn_6
● NVO	1:11	---	SNVT_volt_f	---	nvoAnalogIn_7
● NVO	1:12	---	SNVT_volt_f	---	nvoAnalogIn_8
● NVO	1:13	---	SNVT_volt_f	---	nvoAnalogIn_9
● NVO	1:14	---	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 BACnet objects until the NV List is finalized.

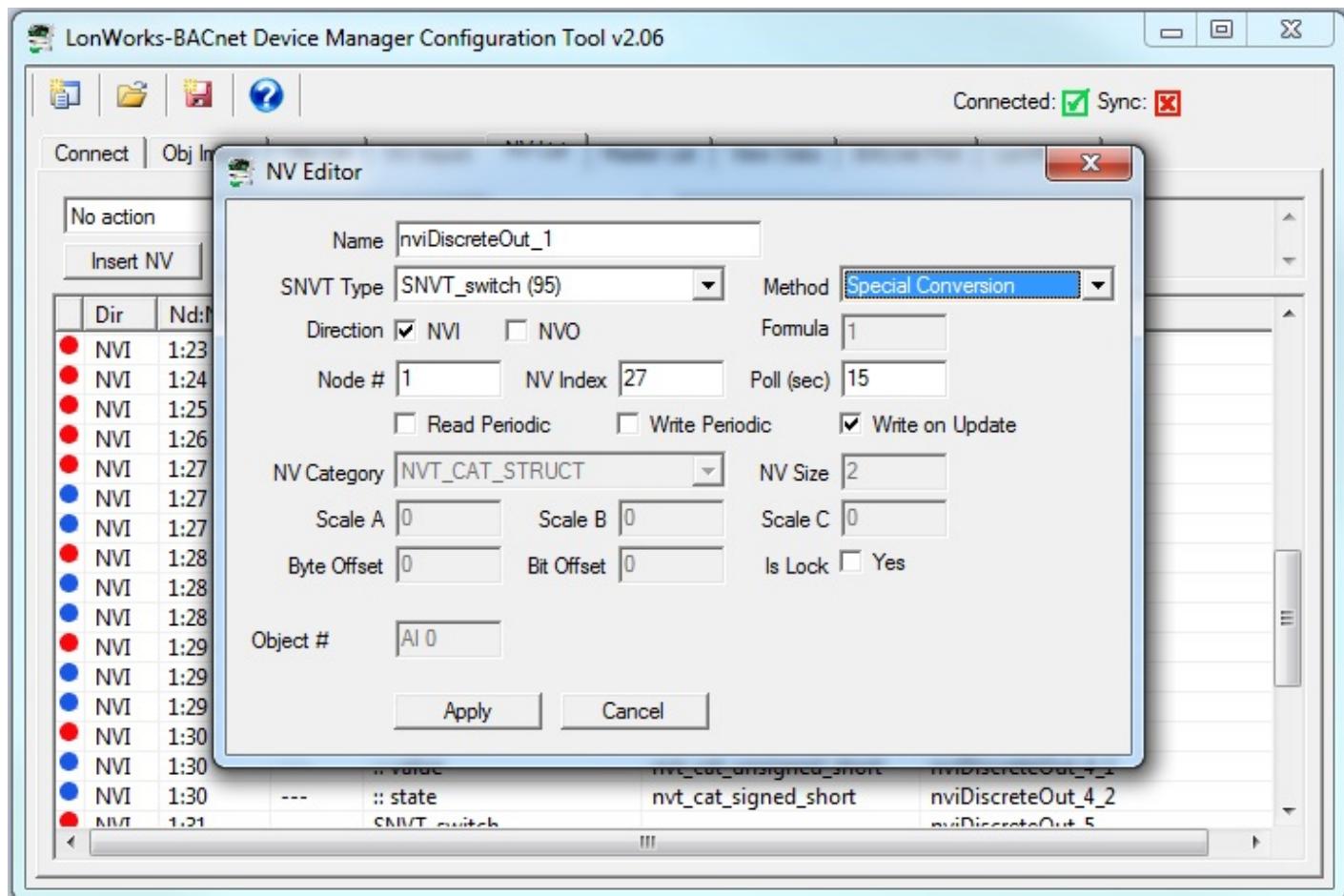
You have a few options here, and these are described in more detail below. You may add network variables, delete them, or change what type they are. If they are structured, they will be automatically expanded into a list of all of their fields. If you add a structured NV which is not a standard LonMark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

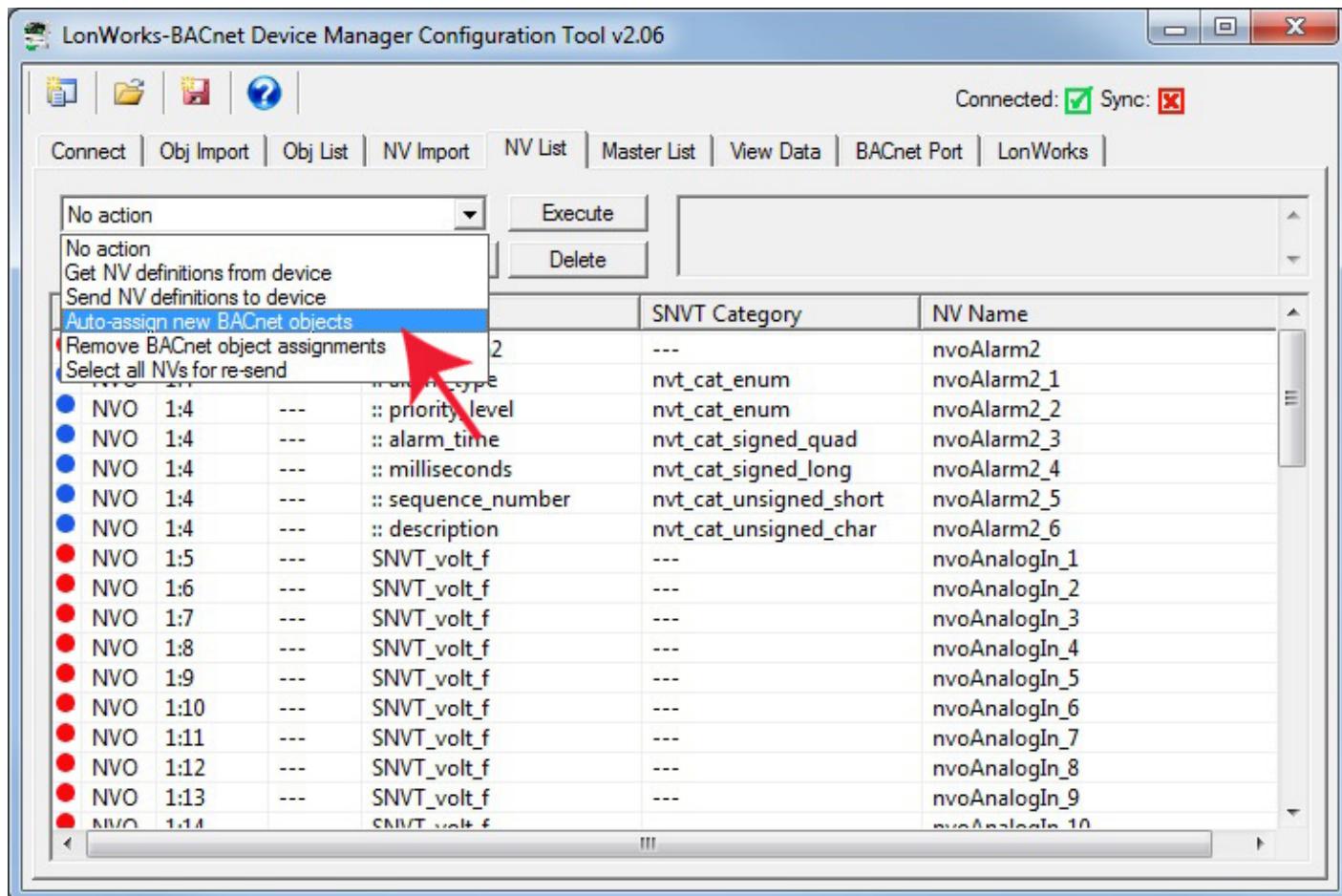
The screenshot shows the 'LonWorks-BACnet Device Manager Configuration Tool v2.06' window. The 'NV List' tab is active. The interface includes a toolbar with icons for file operations, a status bar indicating 'Connected' and 'Sync' status, and a menu bar with 'Connect', 'Obj Import', 'Obj List', 'NV Import', 'NV List' (selected), 'Master List', 'View Data', 'BACnet Port', and 'LonWorks'. Below the menu is a toolbar with buttons for 'No action', 'Execute', 'Insert NV', 'Append NV', 'Add Field', and 'Delete'. The main area is a table titled 'NV List' with columns: Dir, Nd:Nv, Loc ..., SNVT Type, SNVT Category, and NV Name. The table contains multiple rows, with the 11th row (NVI 1:27) highlighted in blue and the 'SNVT Type' cell containing 'SNVT_switch' selected. A large red arrow points to this selected cell. The table scroll bars are visible on the right side.

	Dir	Nd:Nv	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVI	1:23	---	SNVT_count_f	---	nviAnalogOut_1
●	NVI	1:24	---	SNVT_count_f	---	nviAnalogOut_2
●	NVI	1:25	---	SNVT_count_f	---	nviAnalogOut_3
●	NVI	1:26	---	SNVT_count_f	---	nviAnalogOut_4
●	NVI	1:27	---	SNVT_switch	---	nviDiscreteOut_1
●	NVI	1:27	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_1_1
●	NVI	1:27	---	:: state	nvt_cat_signed_short	nviDiscreteOut_1_2
●	NVI	1:28	---	SNVT_switch	---	nviDiscreteOut_2
●	NVI	1:28	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_2_1
●	NVI	1:28	---	:: state	nvt_cat_signed_short	nviDiscreteOut_2_2
●	NVI	1:29	---	SNVT_switch	---	nviDiscreteOut_3
●	NVI	1:29	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_3_1
●	NVI	1:29	---	:: state	nvt_cat_signed_short	nviDiscreteOut_3_2
●	NVI	1:30	---	SNVT_switch	---	nviDiscreteOut_4
●	NVI	1:30	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_4_1
●	NVI	1:30	---	:: state	nvt_cat_signed_short	nviDiscreteOut_4_2
●	NVI	1:31	---	SNVT_switch	---	nviDiscreteOut_5

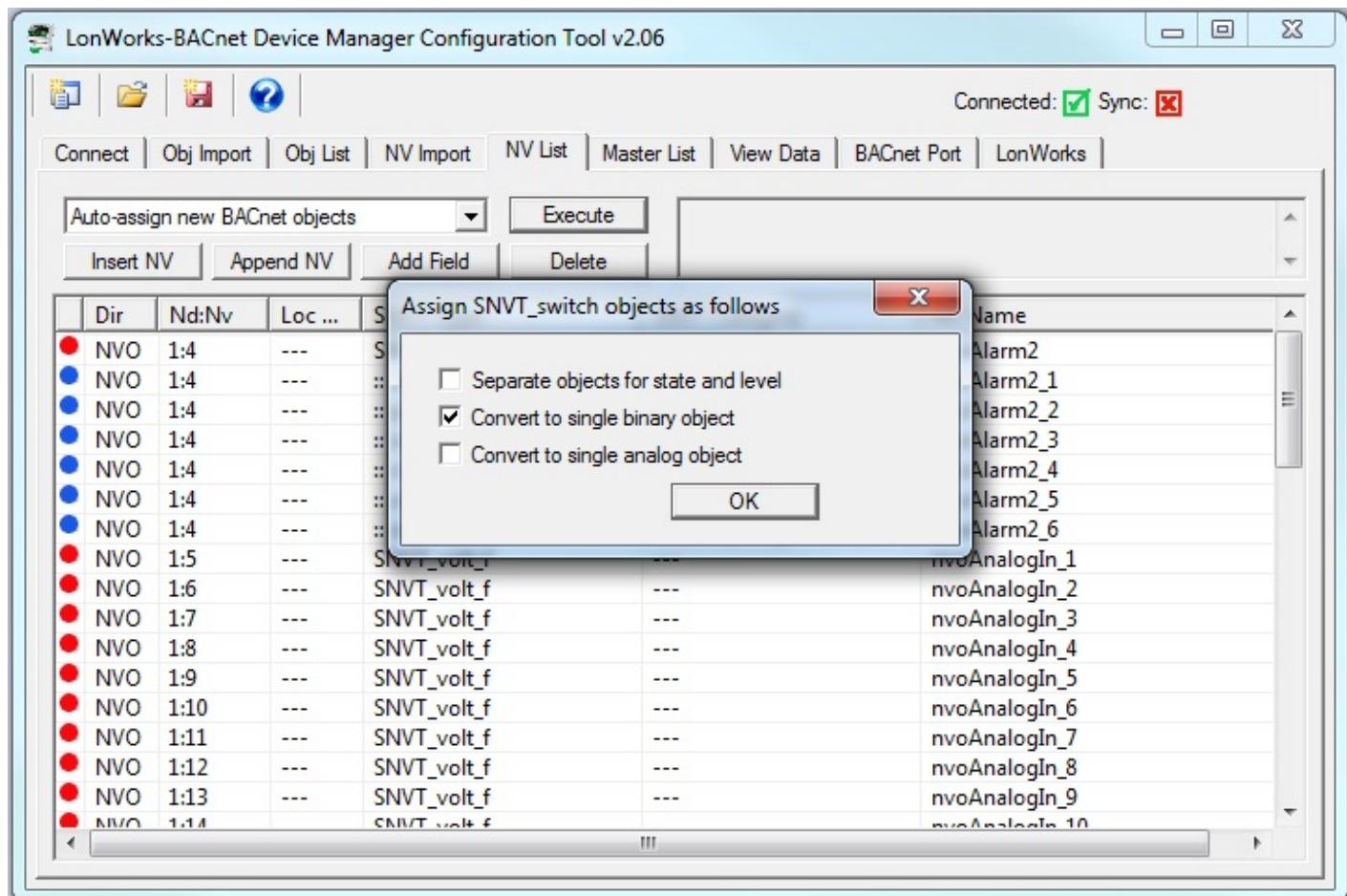
Upon double clicking a network variable in the NV List, the NV Editor dialog will appear. The SNVT_switch example is illustrated here. Your selection of whether to use the special conversion will decide whether SNVT_switch maps to one or two BACnet objects, and therefore whether SNVT_switch maps to one 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.



Once the list of network variables is acceptable to you, select and execute "Auto-assign new BACnet objects".



When auto-assigning objects, a dialog will pop up asking about your preference for treatment of any SNVT_switch variables that may be in the list. LonWorks treatment of switches assumes a dimmer type switch, and the SNVT_switch variable has both a state and a level, both of which must be provided on the LonWorks side. You have 3 options for how you will treat this on the BACnet side: (a) You can assign two objects, one each for state and level; (b) You can assign an analog object which is then anticipating a value between 0 and 100 (percent implied); or (c) You can assign a binary object, in which case the gateway automatically translates 'inactive' into the 'off' state with 0% level, and 'active' into the 'on' state with 100% level.



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

Dir	Nd:Nv	Loc ...	SNVT Type	SNVT Category	NV Name
● NVO	1:4	---	SNVT_alarm2	---	nvoAlarm2
● NVO	1:4	AI1	:: alarm_type	nvt_cat_enum	nvoAlarm2_1
● NVO	1:4	AI2	:: priority_level	nvt_cat_enum	nvoAlarm2_2
● NVO	1:4	AI3	:: alarm_time	nvt_cat_signed_quad	nvoAlarm2_3
● NVO	1:4	AI4	:: milliseconds	nvt_cat_signed_long	nvoAlarm2_4
● NVO	1:4	AI5	:: sequence_number	nvt_cat_unsigned_short	nvoAlarm2_5
● NVO	1:4	AI6	:: description	nvt_cat_unsigned_char	nvoAlarm2_6
● NVO	1:5	AI7	SNVT_volt_f	---	nvoAnalogIn_1
● NVO	1:6	AI8	SNVT_volt_f	---	nvoAnalogIn_2
● NVO	1:7	AI9	SNVT_volt_f	---	nvoAnalogIn_3
● NVO	1:8	AI10	SNVT_volt_f	---	nvoAnalogIn_4
● NVO	1:9	AI11	SNVT_volt_f	---	nvoAnalogIn_5
● NVO	1:10	AI12	SNVT_volt_f	---	nvoAnalogIn_6
● NVO	1:11	AI13	SNVT_volt_f	---	nvoAnalogIn_7
● NVO	1:12	AI14	SNVT_volt_f	---	nvoAnalogIn_8
● NVO	1:13	AI15	SNVT_volt_f	---	nvoAnalogIn_9
● NIO	1:14	AI16	SNVT_volt_f	---	nvoAnalogIn_10

The example configuration below shows that SNVT_switch variables have been converted to a single Binary object.

LonWorks-BACnet Device Manager Configuration Tool v2.06

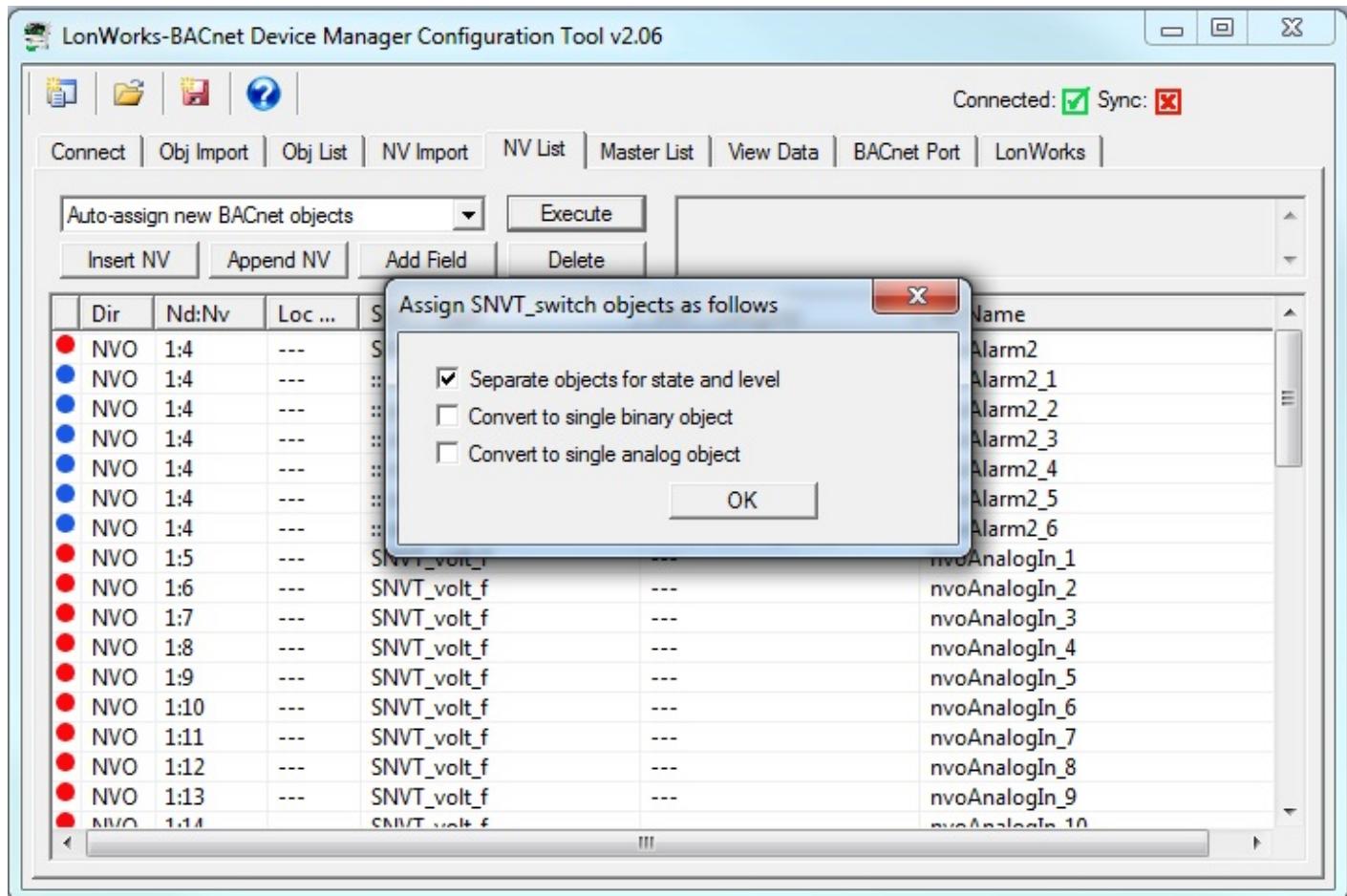
Connected: Sync:

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	Nd:Nv	Loc ...	SNVT Type	SNVT Category	NV Name
● NVI	1:23	AO 1	SNVT_count_f	---	nviAnalogOut_1
● NVI	1:24	AO 2	SNVT_count_f	---	nviAnalogOut_2
● NVI	1:25	AO 3	SNVT_count_f	---	nviAnalogOut_3
● NVI	1:26	AO 4	SNVT_count_f	---	nviAnalogOut_4
● NVI	1:27	BO 1	SNVT_switch	---	nviDiscreteOut_1
● NVI	1:27	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_1_1
● NVI	1:27	---	:: state	nvt_cat_signed_short	nviDiscreteOut_1_2
● NVI	1:28	BO 2	SNVT_switch	---	nviDiscreteOut_2
● NVI	1:28	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_2_1
● NVI	1:28	---	:: state	nvt_cat_signed_short	nviDiscreteOut_2_2
● NVI	1:29	BO 3	SNVT_switch	---	nviDiscreteOut_3
● NVI	1:29	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_3_1
● NVI	1:29	---	:: state	nvt_cat_signed_short	nviDiscreteOut_3_2
● NVI	1:30	BO 4	SNVT_switch	---	nviDiscreteOut_4
● NVI	1:30	---	:: value	nvt_cat_unsigned_short	nviDiscreteOut_4_1
● NVI	1:30	---	:: state	nvt_cat_signed_short	nviDiscreteOut_4_2
● NVI	1:31	BO 5	SNVT_switch	---	nviDiscreteOut_5

You might also select 'Separate objects for state and level'.

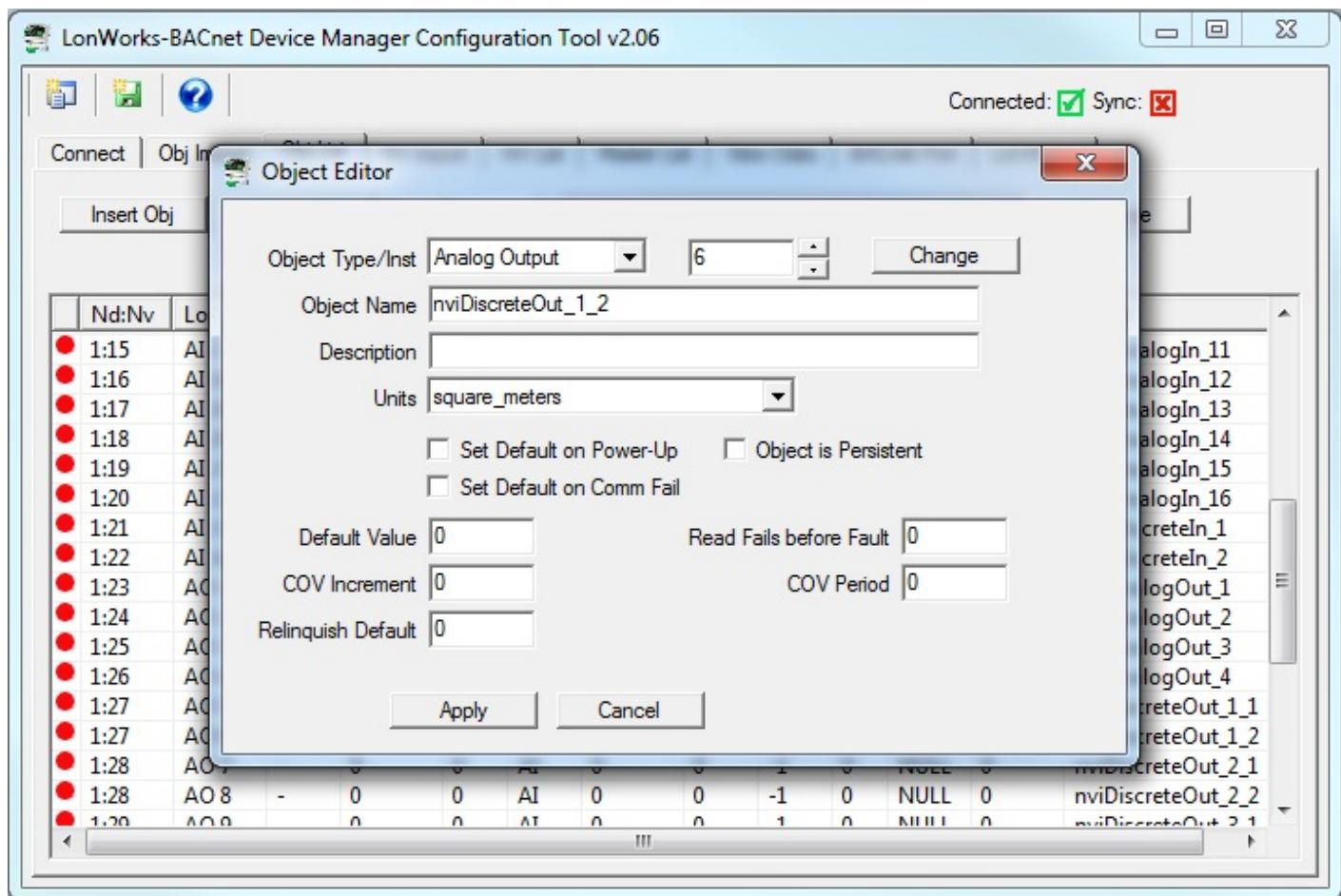


The resulting automatic assignment of objects will then appear as follows.

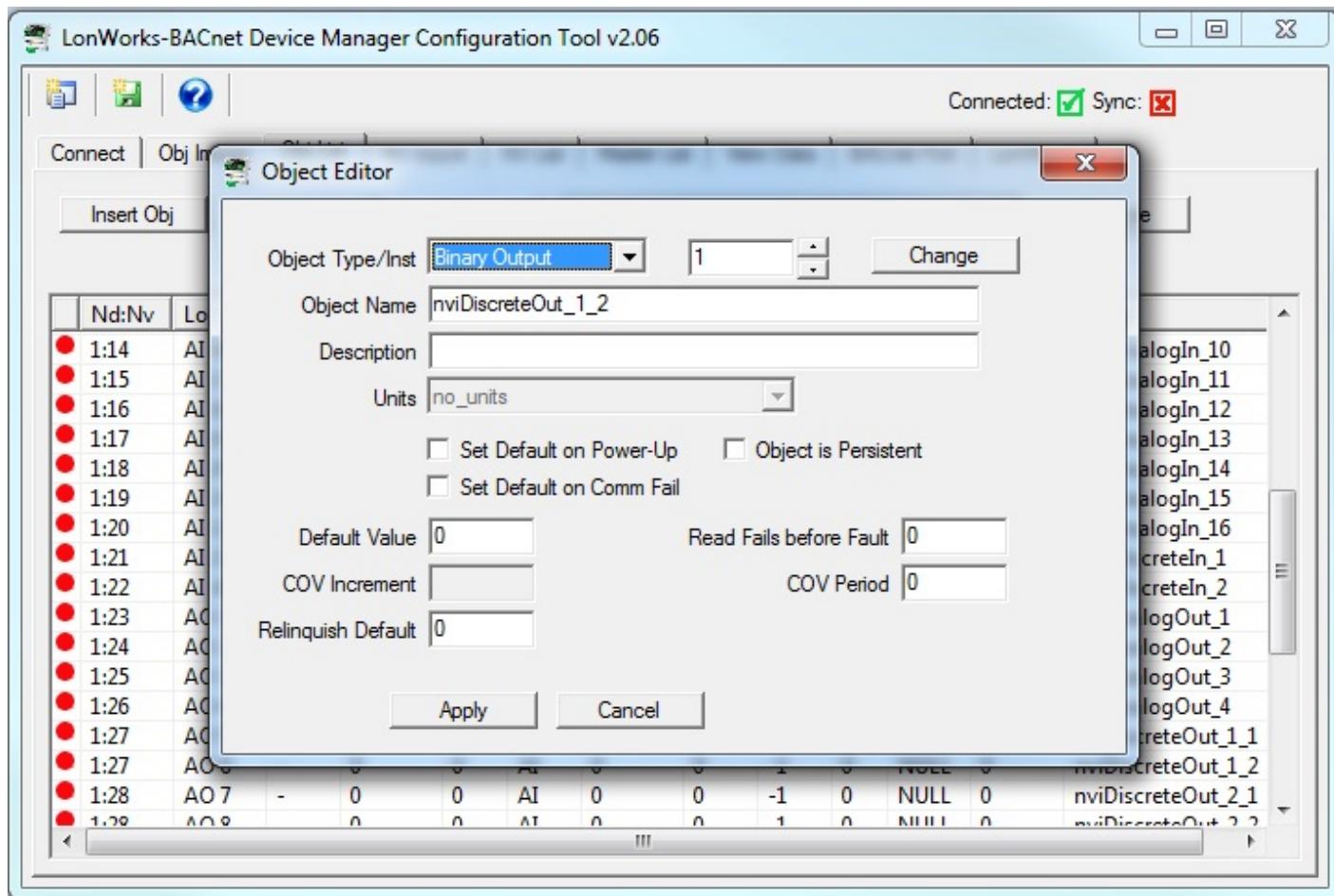
The screenshot shows the LonWorks-BACnet Device Manager Configuration Tool v2.06. The NV List tab is selected. The interface includes a toolbar with icons for file operations, a status bar indicating 'Connected' and 'Sync' status, and a menu bar with options like Connect, Obj Import, Obj List, NV Import, NV List, Master List, View Data, BACnet Port, and LonWorks.

Dir	Nd:Nv	Loc ...	SNVT Type	SNVT Category	NV Name
● NVI	1:23	AO 1	SNVT_count_f	---	nviAnalogOut_1
● NVI	1:24	AO 2	SNVT_count_f	---	nviAnalogOut_2
● NVI	1:25	AO 3	SNVT_count_f	---	nviAnalogOut_3
● NVI	1:26	AO 4	SNVT_count_f	---	nviAnalogOut_4
● NVI	1:27	---	SNVT_switch	---	nviDiscreteOut_1
● NVI	1:27	AO 5	:: value	nvt_cat_unsigned_short	nviDiscreteOut_1_1
● NVI	1:27	AO 6	:: state	nvt_cat_signed_short	nviDiscreteOut_1_2
● NVI	1:28	---	SNVT_switch	---	nviDiscreteOut_2
● NVI	1:28	AO 7	:: value	nvt_cat_unsigned_short	nviDiscreteOut_2_1
● NVI	1:28	AO 8	:: state	nvt_cat_signed_short	nviDiscreteOut_2_2
● NVI	1:29	---	SNVT_switch	---	nviDiscreteOut_3
● NVI	1:29	AO 9	:: value	nvt_cat_unsigned_short	nviDiscreteOut_3_1
● NVI	1:29	AO 10	:: state	nvt_cat_signed_short	nviDiscreteOut_3_2
● NVI	1:30	---	SNVT_switch	---	nviDiscreteOut_4
● NVI	1:30	AO 11	:: value	nvt_cat_unsigned_short	nviDiscreteOut_4_1
● NVI	1:30	AO 12	:: state	nvt_cat_signed_short	nviDiscreteOut_4_2
● NVI	1:31		CNV/T switch		nviDiscreteOut_5

You may decide that you would rather have Binary objects for the state portion of the SNVT_switch. To change object type, go to the Master List instead. Double click on the object portion of the line to be changed. This will open the Object Editor. Now select a different object type from the list.



We are choosing to change from Analog Output to Binary Output. When this change is made, the tool searches for the first instance of an unused Binary Output object, and places that instance number in the window next to object type. If you wish to change instance number you may, otherwise simply click Apply at this point.

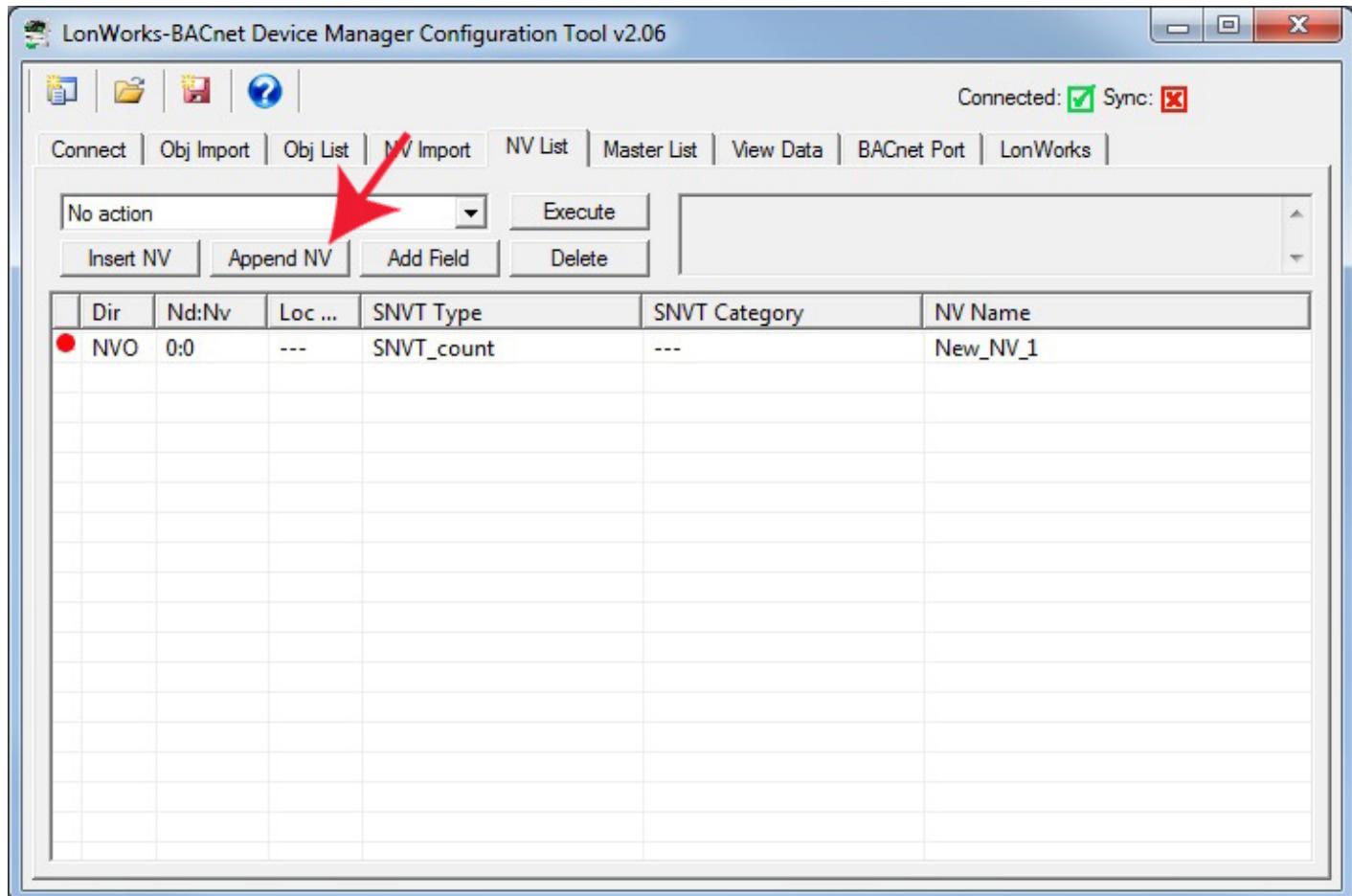


8.2 Building Configuration Manually

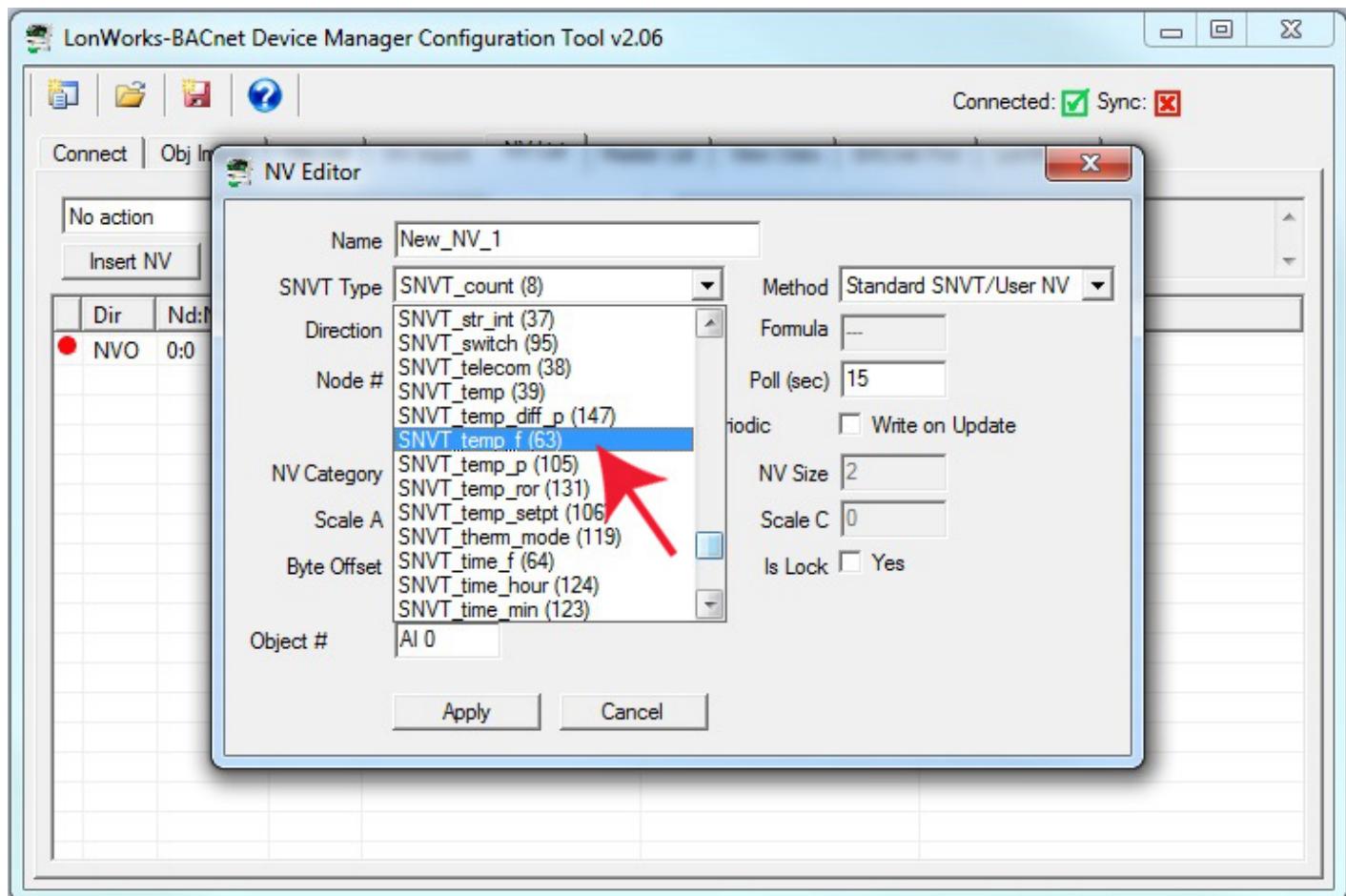
It is not required that you start with either a CSV file or an XIF file. You can use the configuration tool to start with a blank slate and build your configuration from scratch.

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.

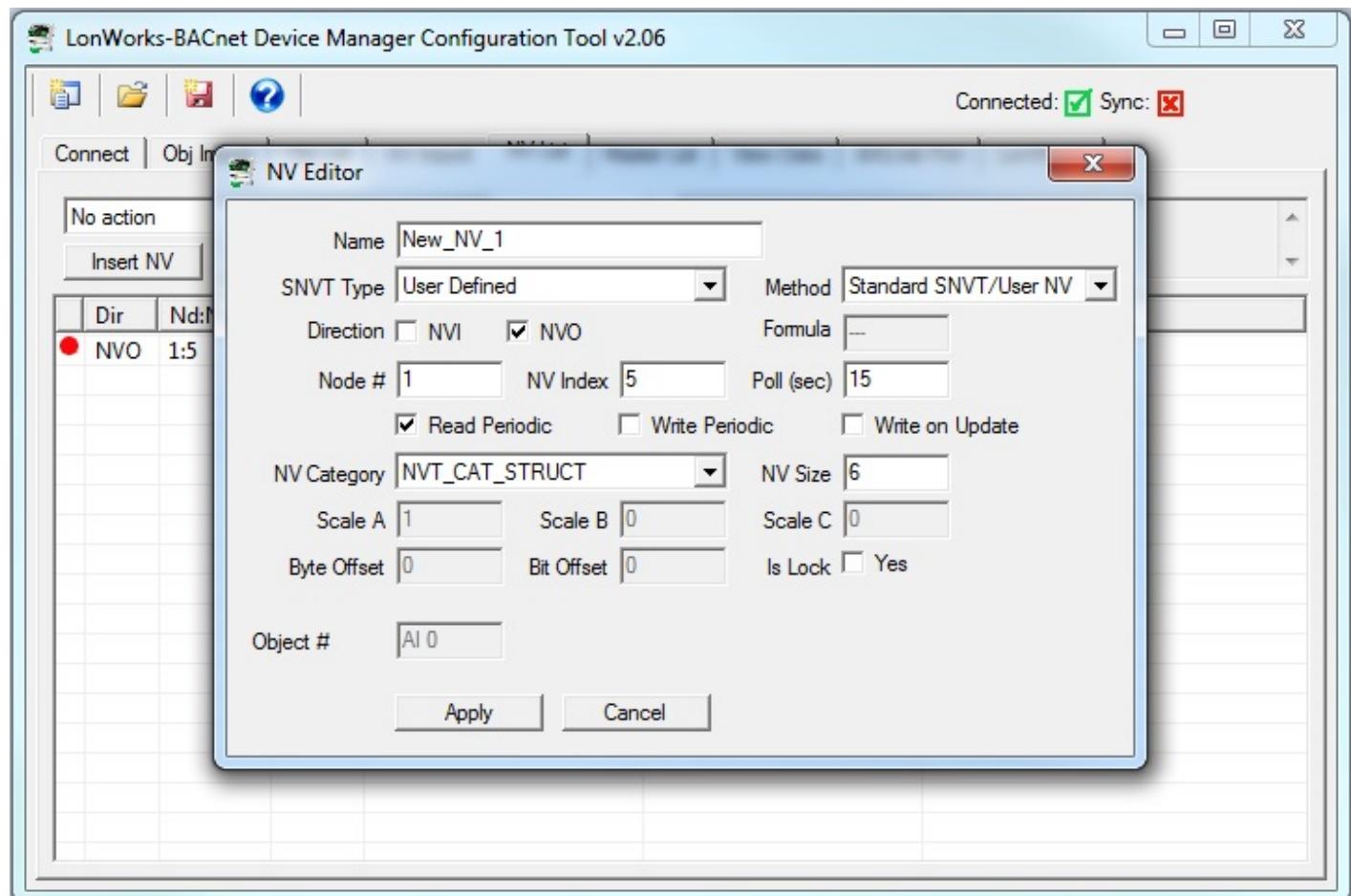
For each network variable you wish to add, click Append NV to add one at the end of the list, or Insert NV to add one immediately before the selected line on the list (if any).



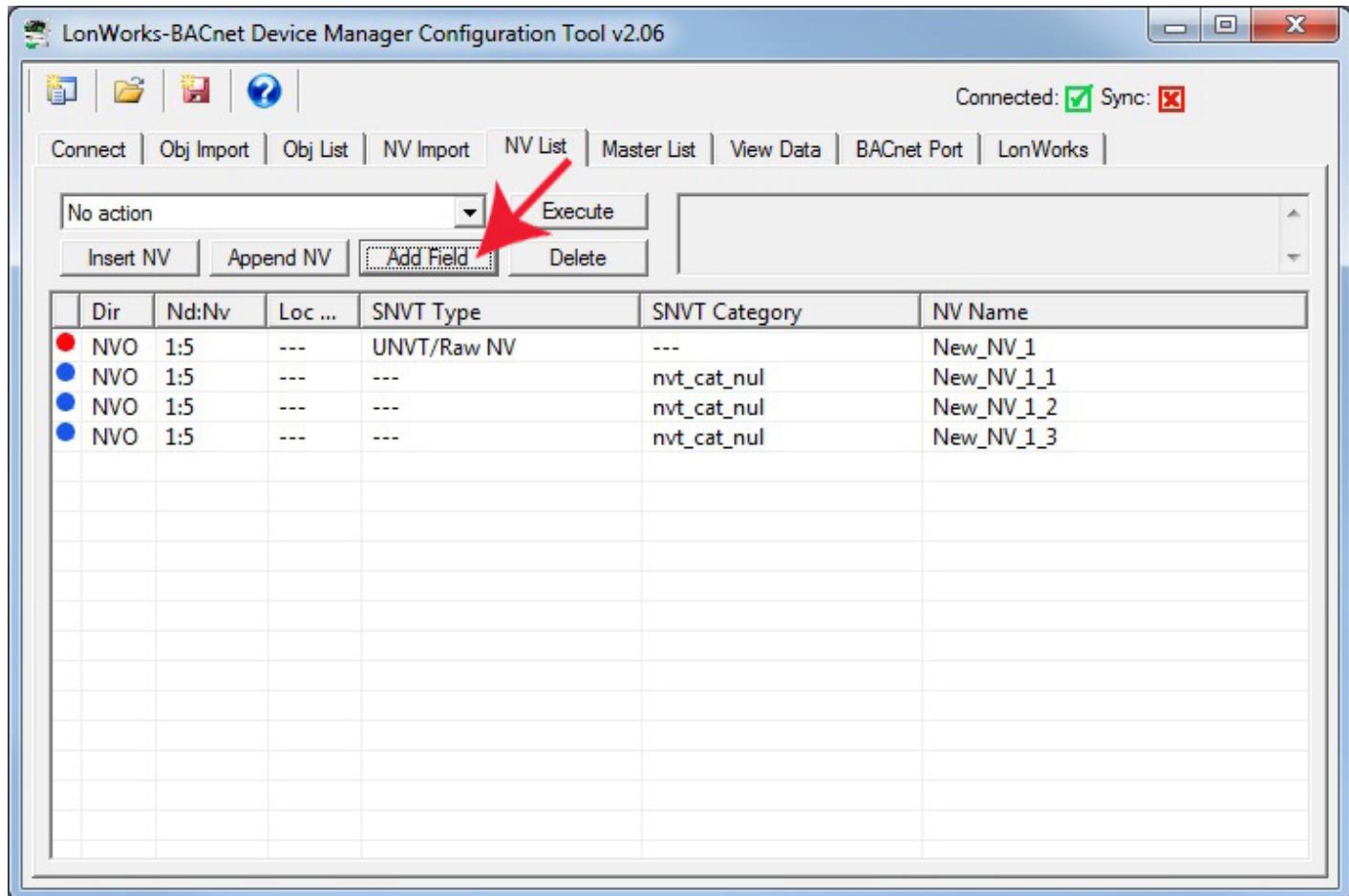
The newly inserted NV will default to SNVT_count. If you wish to change it, double click on the line to be changed, and the NV Editor dialog will appear. In the example that follows, we will complicate things to the maximum by creating a user defined structured network variable.



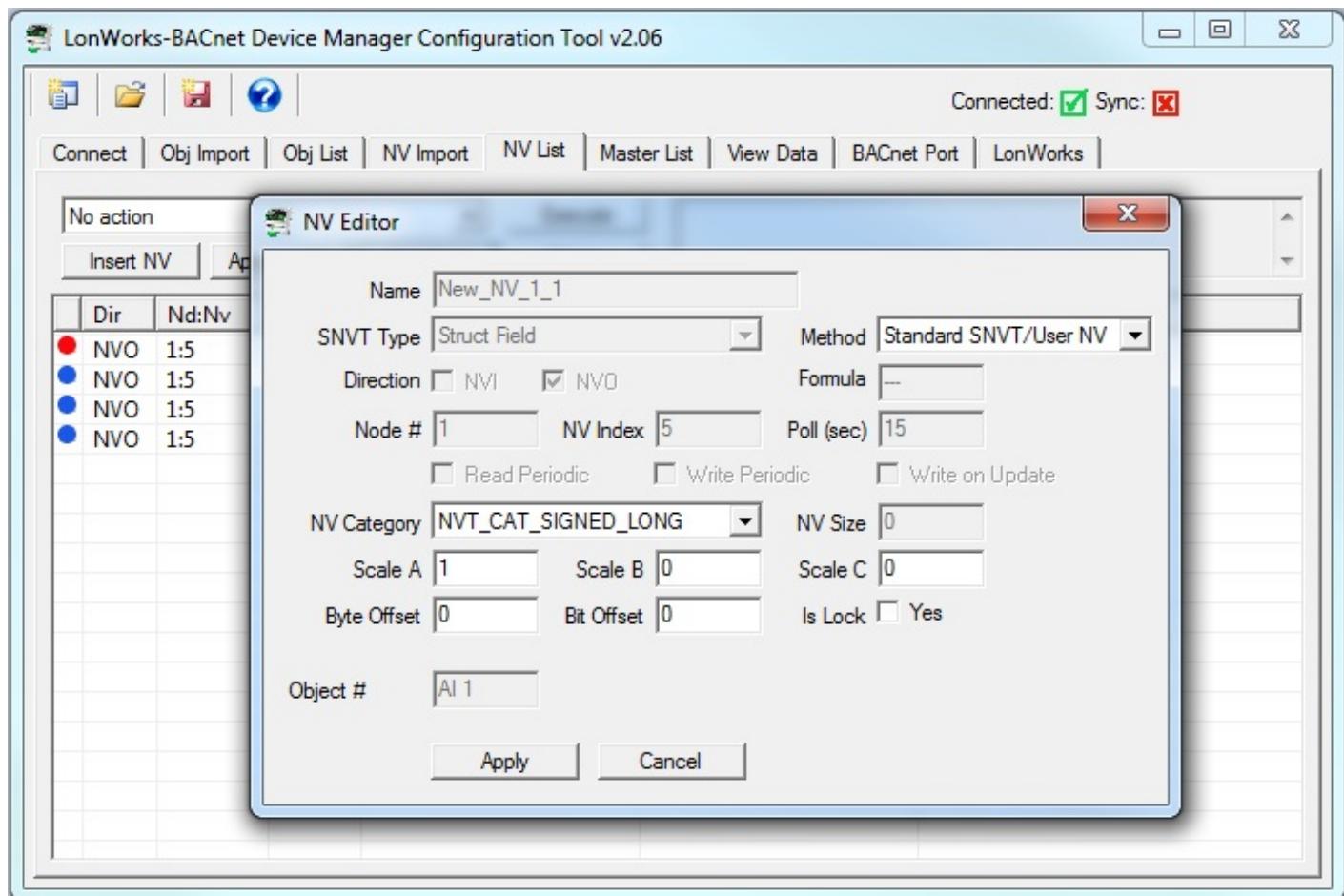
When creating a structure, the NV needs to be set to NV Category NVT_CAT_STRUCT and the NV Size needs to be set to the number of bytes that make up this variable in the LonWorks device.



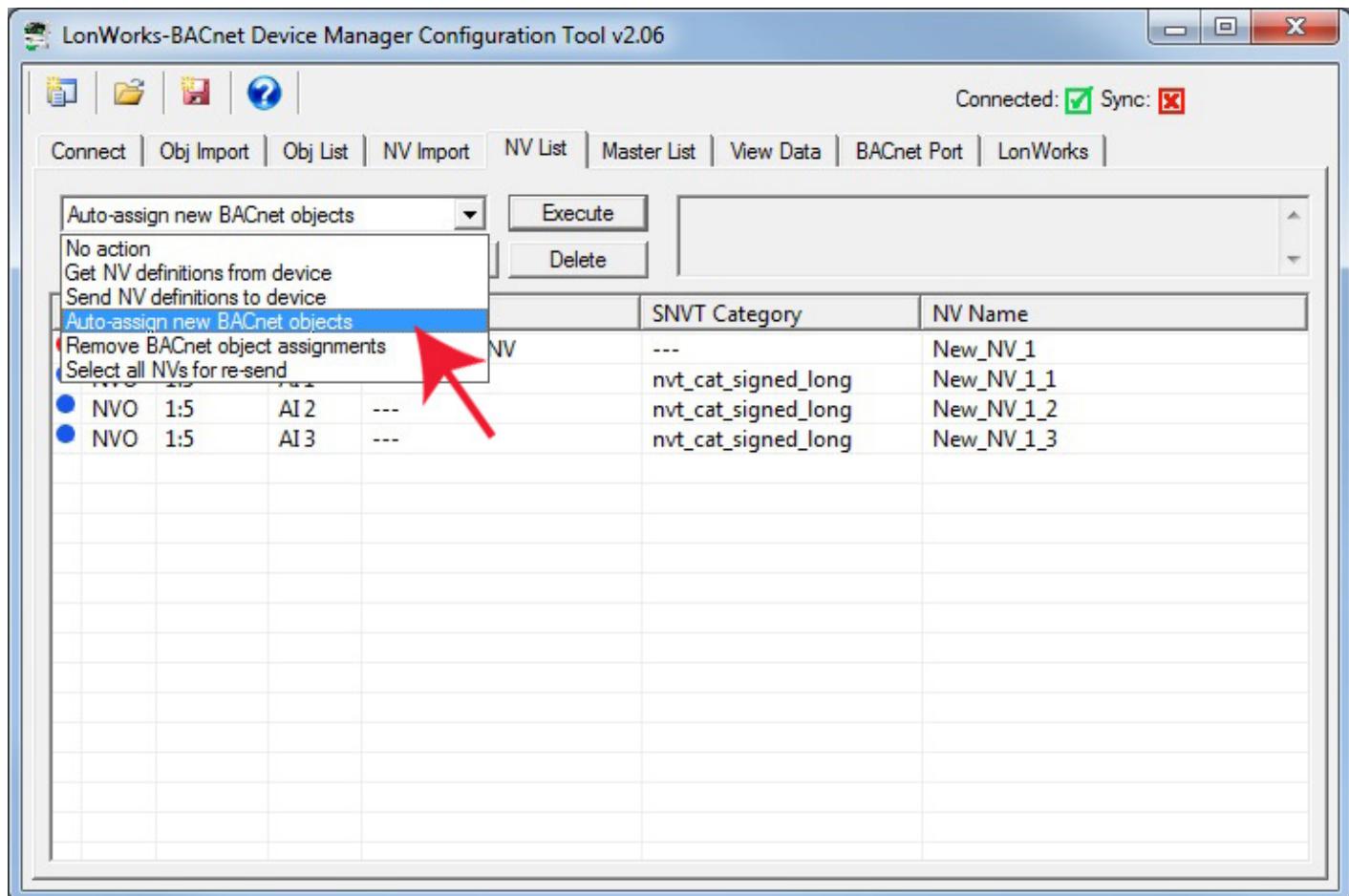
After defining the NV as a structure, you next need to add fields to the structure. Do this by clicking Add Field.



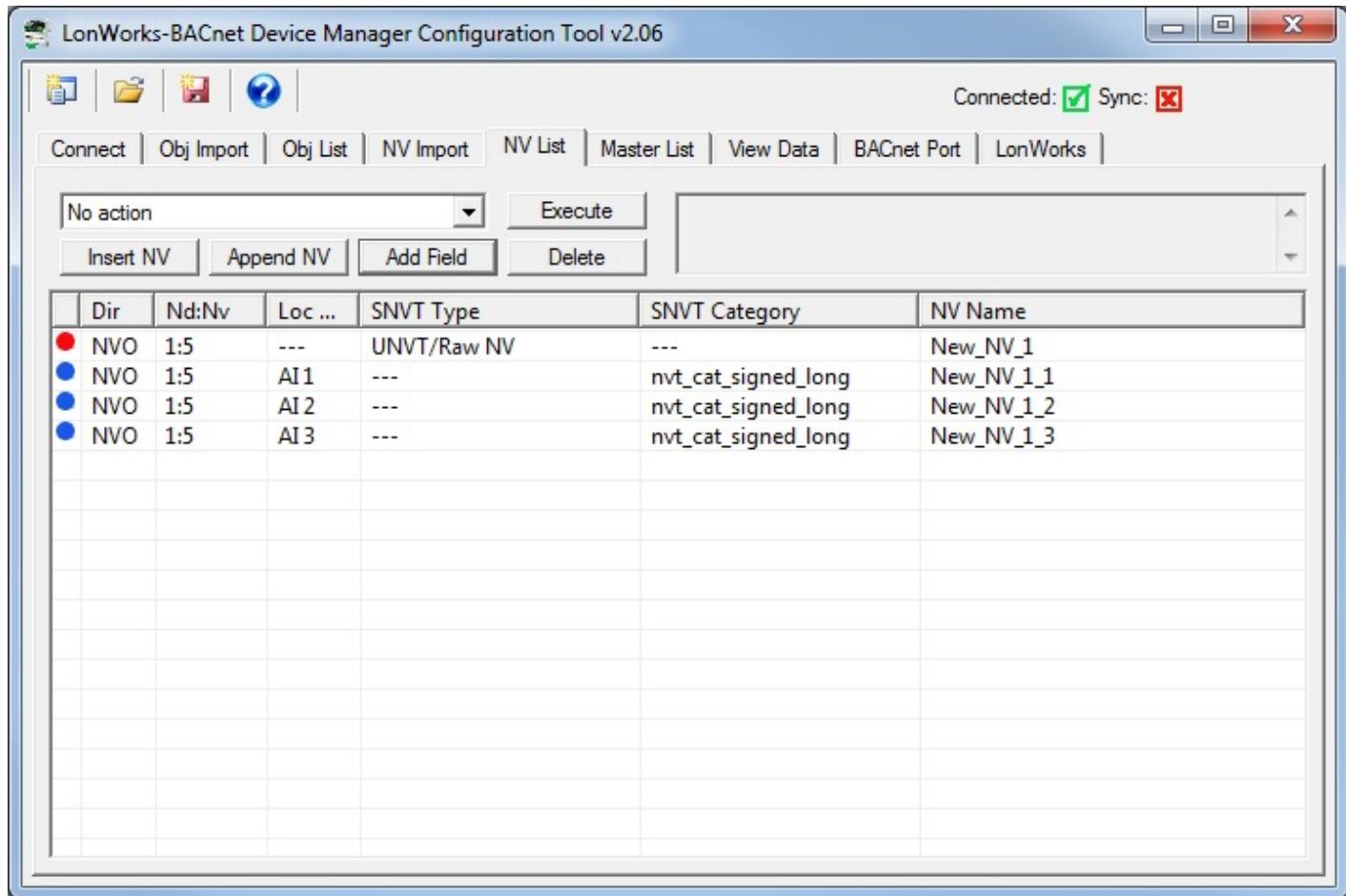
Next, edit each field by double clicking on that line. Specify the data type for the field. It is also important to specify the Byte Offset in the structure. An offset of zero means this field occupies the first byte(s) of the structure. You also need to enter the scale values. These follow the LonMark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



In addition to setting up the Network Variable itself, you need to assign BACnet objects. These can be manually entered, or automatically assigned by the configuration tool. Once assigned, you need to go to the Master List page, click on the object portion of the Master List entry, and change type or instance there.

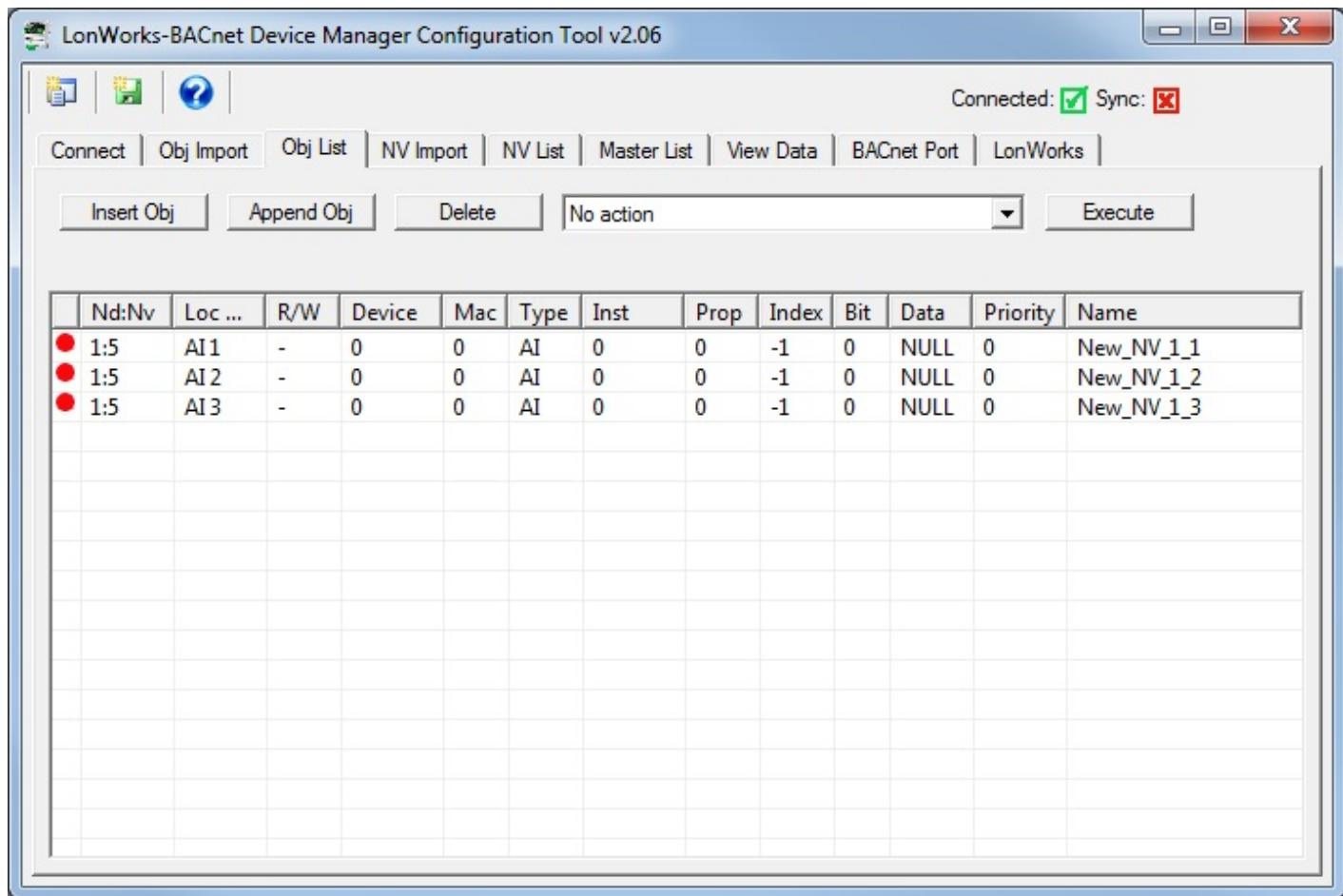


Following configuration of the NV and its fields, your NV List will appear as follows for this first single, user defined, structured NV that will map to three BACnet objects.



Following BACnet object assignment, the objects will also appear on the Object List.

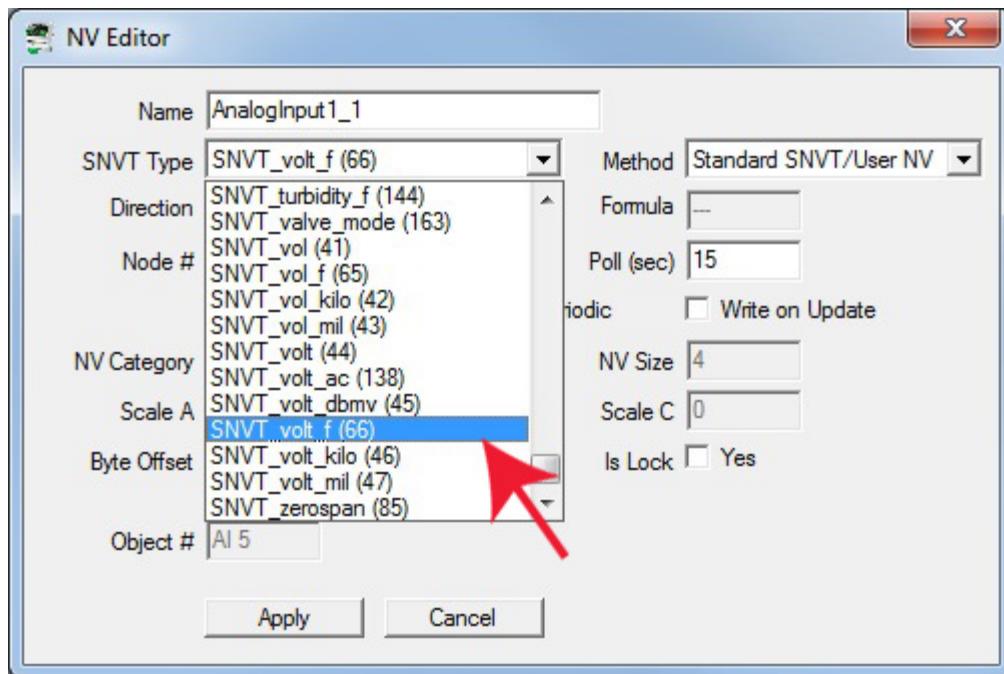
IMPORTANT: If you have manually assigned BACnet objects using the NV Editor dialog, you must still select 'Auto-assign new BACnet objects' from the list on the NV List page and click Execute. This will not find any new objects to assign, but the important thing it does do is make the connections between NV List and Object List. Until you do the auto-assign process even for manually assigned objects, they will not appear in the object list on the Obj List page.



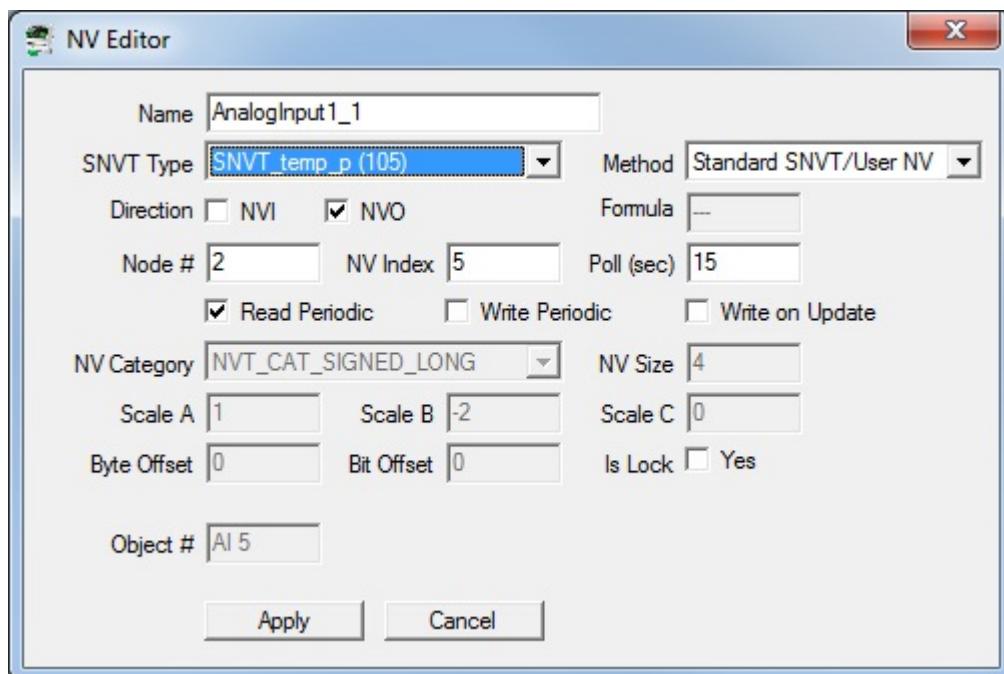
8.3 Using the NV Editor

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.

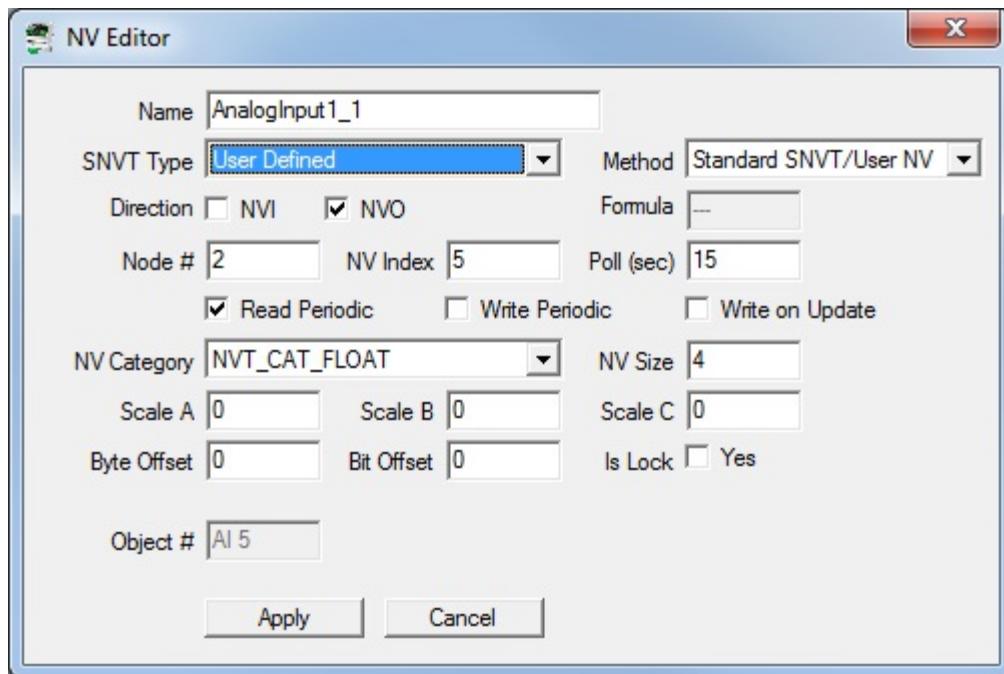
The following example illustrates the most simple of the NV Editor operations, namely just selecting a different standard SNVT type from the list of SNVT Types. When a standard LonMark SNVT type is selected, most of the configuration parameters will be fixed by LonMark and set to read-only in the editor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



The following is an example of a standard SNVT selection:

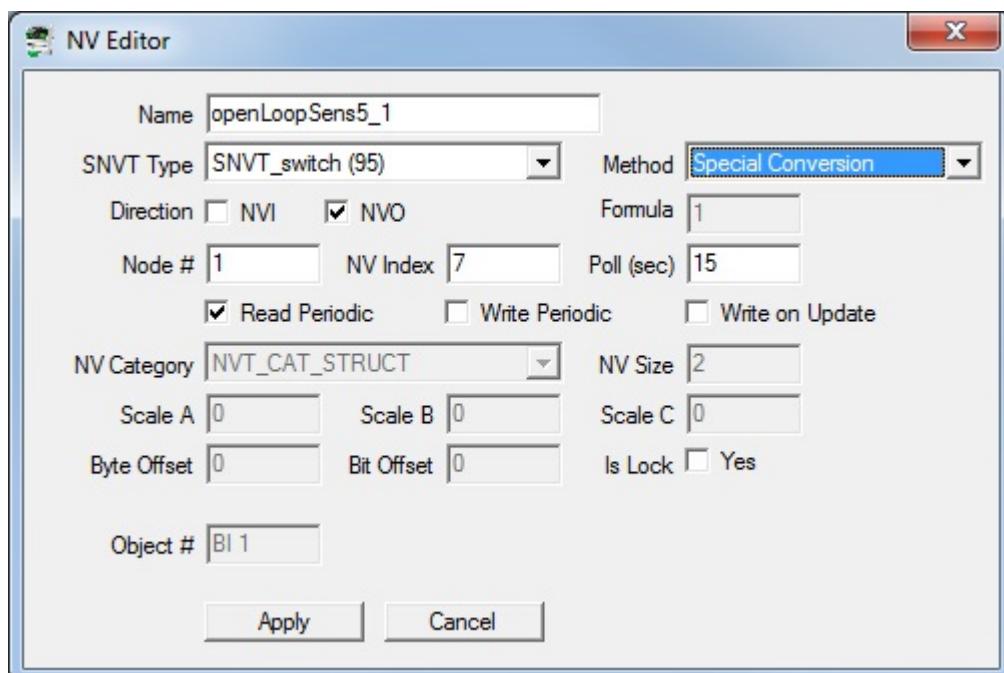


If you were to create a 'user defined' NV that provided the exact same results as SNVT_temp, the NV Editor dialog would appear as follows:

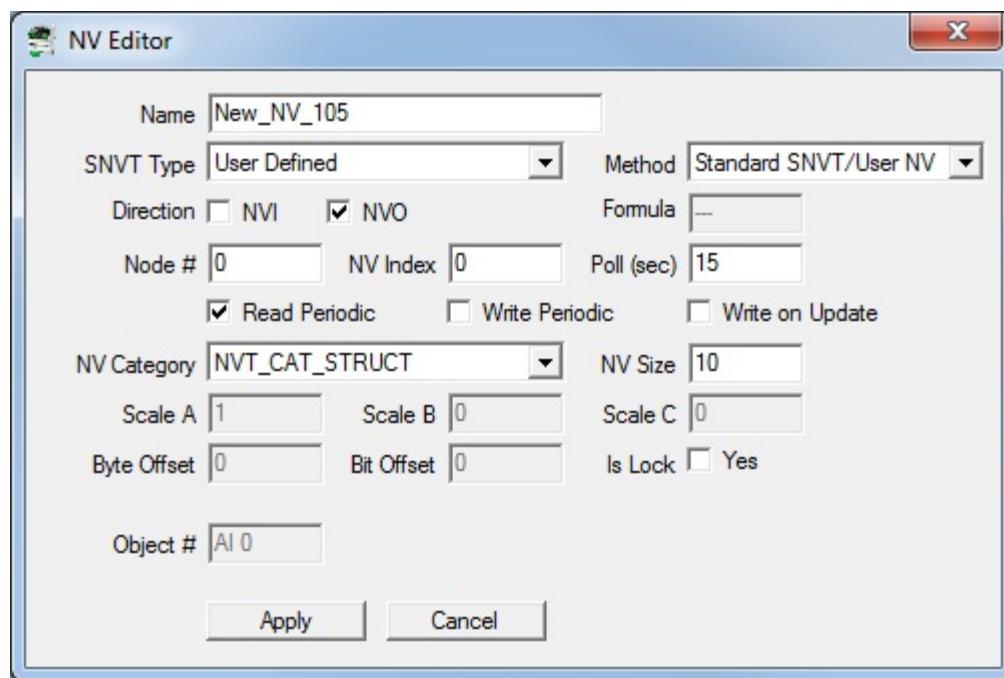


The SNVT_switch example is illustrated below. The LonMark definition of a "switch" actually contains two elements of data, a state and a level (as would be used for a dimmer switch). You cannot correctly control anything via a SNVT_switch without properly dealing with both elements of data. In BACnet 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. The special conversion is also extended to check specifically for Binary objects, and convert inactive and active states to 0% and 100% levels for SNVT_switch respectively.

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 BACnet objects.



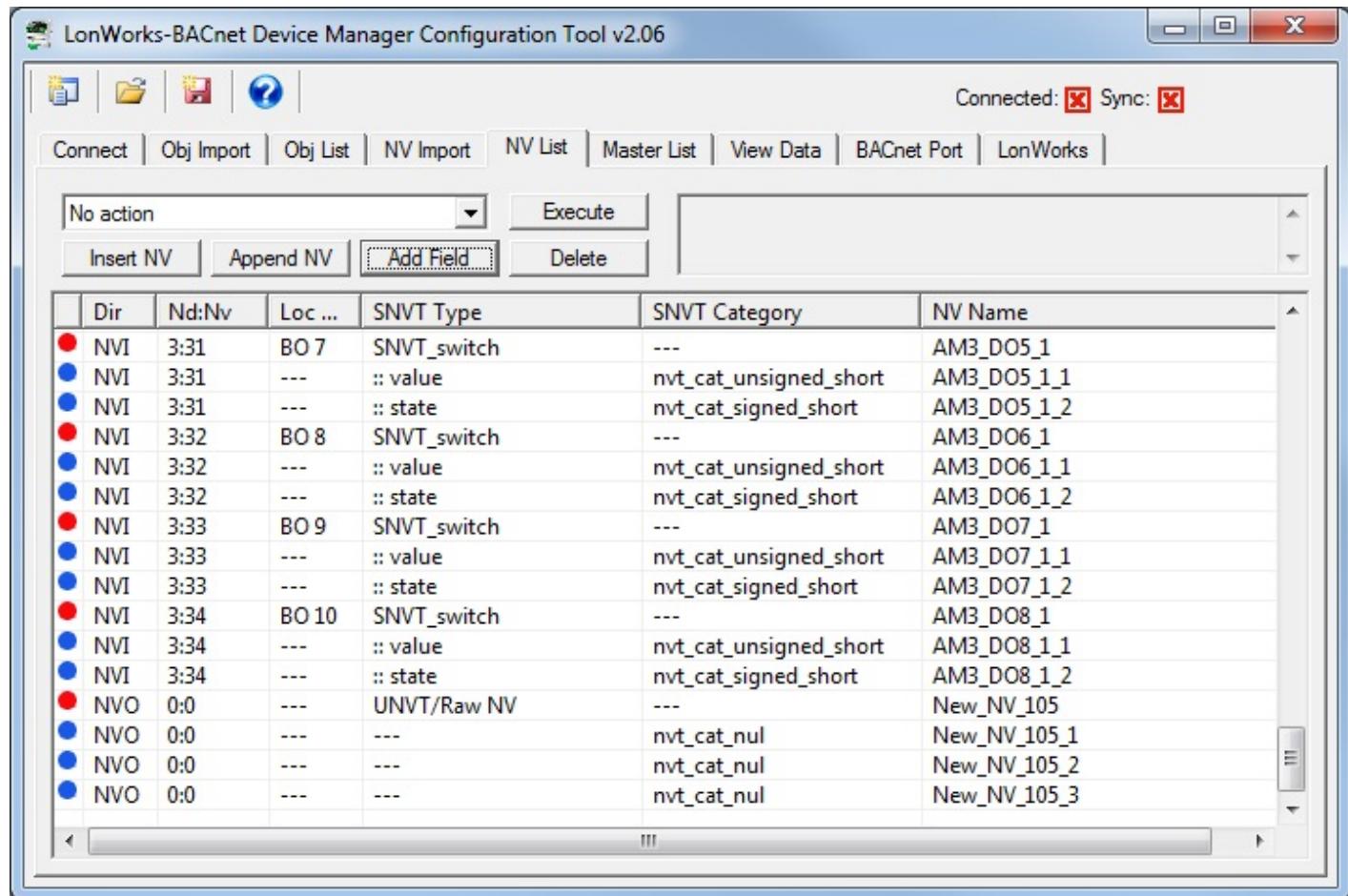
Creating a user defined structured network variable is illustrated below. The parent NV must have SNVT type set to User Defined. The NV Category must be NVT_CAT_STRUCT and the NV Size must be set to the total number of bytes of data that this NV will occupy, including all fields of the structure.



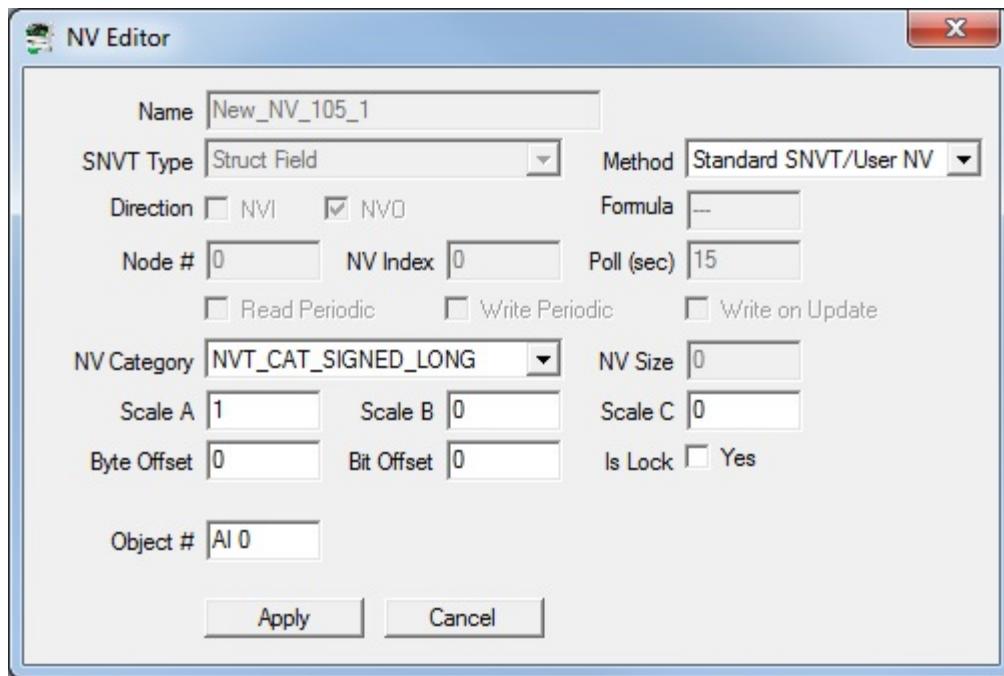
Once a user defined structured Network Variable is created, you need to add data fields to the structure by clicking Add Field (after selecting the UNVT by single clicking on it in the list).

Dir	Nd:Nv	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVI	3:30	BO 6	SNVT_switch	---
●	NVI	3:30	---	:: value	nvt_cat_unsigned_short
●	NVI	3:30	---	:: state	nvt_cat_signed_short
●	NVI	3:31	BO 7	SNVT_switch	---
●	NVI	3:31	---	:: value	nvt_cat_unsigned_short
●	NVI	3:31	---	:: state	nvt_cat_signed_short
●	NVI	3:32	BO 8	SNVT_switch	---
●	NVI	3:32	---	:: value	nvt_cat_unsigned_short
●	NVI	3:32	---	:: state	nvt_cat_signed_short
●	NVI	3:33	BO 9	SNVT_switch	---
●	NVI	3:33	---	:: value	nvt_cat_unsigned_short
●	NVI	3:33	---	:: state	nvt_cat_signed_short
●	NVI	3:34	BO 10	SNVT_switch	---
●	NVI	3:34	---	:: value	nvt_cat_unsigned_short
●	NVI	3:34	---	:: state	nvt_cat_signed_short
■	NVO	0:0	---	UNVT/Raw NV	New_NV_105

The unconfigured, empty fields of the structure will look something like the following example.

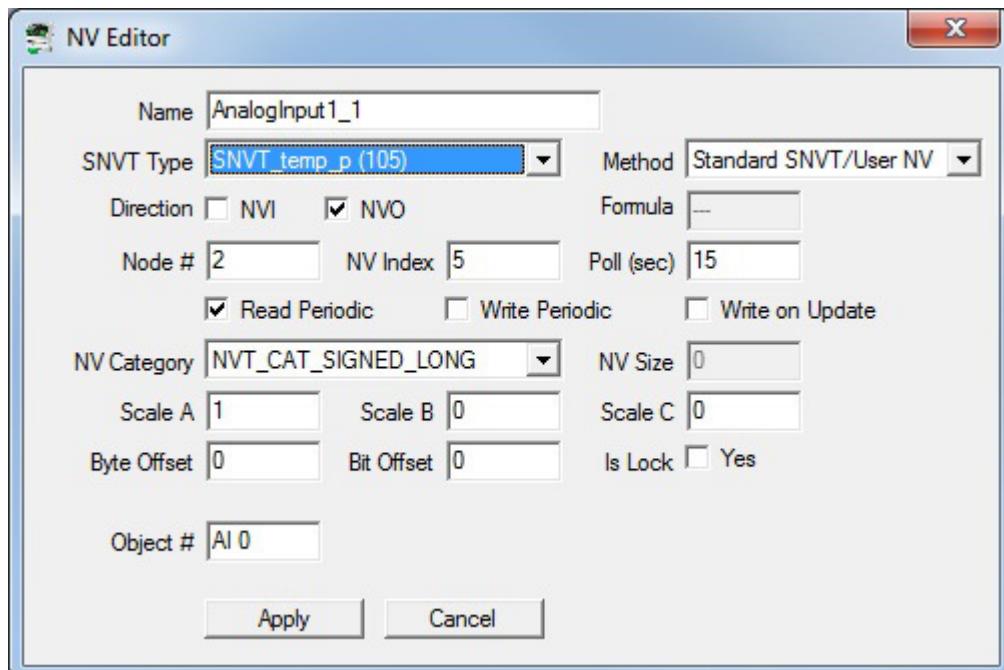


For each field of the structured NV, select a standard data type for NV Category. The first field will have a byte offset of zero. But all subsequent fields will need to have a non-zero offset so that the data is taken from the right place in the structure. When the network variable (NV) is read from the LonWorks device, or sent to 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.



Special conversions available for certain SNVT's are listed below. The conversion 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 initially 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 saved in the gateway device just for reference. The name is limited to 20 characters.
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.
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.
Node #	Indicates which node entry in the node table will be polled for this NV mapping.
NV Index	Specifies the NV index in the remote LonWorks device that will be polled. This is usually taken from the XIF file, otherwise must be obtained from the device manufacturer's documentation. This is effectively the 'address' of the variable within the device, and cannot be arbitrary.
Poll (sec)	Polling period in seconds indicates how frequently the gateway will read or periodically write a Network Variable in the remote LonWorks device.
Read/Write	The choice for read/write must match up with the direction. You can read an NVI or NVO. However, you can only write to an NVI in the remote device. Writing can be periodic at the poll rate, or 'on update' meaning it will only be written to when the BACnet object mapped to this NV is written to.
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 $S = a * 10^b * (R + c)$ 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.
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.

8.4 Using NV CSV Files to Build Multi-Device Configuration

The ability to save and re-load NV lists from a CSV file has been added to this configuration tool. If you have used the XIF import from device, it becomes more manageable to import the NV list, clear the gateway, then import more devices. Save each individual device's NV list to a CSV file. Then use a spread sheet program to combine and edit the lists.

The format expected for the NV CSV file is given in Appendix F.

```
NODE,NVINDEX,DIR,SERVICE,SNVT,UNVTSIZE,NAME
1,5,0,R,66,0,AnalogInput1_1
1,6,0,R,66,0,AnalogInput2_1
1,7,0,R,66,0,AnalogInput3_1
1,8,0,R,66,0,AnalogInput4_1
1,9,1,W+,95,0,DiscreteOutp1_1
1,10,1,W+,95,0,DiscreteOutp2_1
2,5,0,R,66,0,AnalogInput1_1
2,6,0,R,66,0,AnalogInput2_1
2,7,0,R,66,0,AnalogInput3_1
2,8,0,R,66,0,AnalogInput4_1
2,9,1,W+,95,0,DiscreteOutp1_1
2,10,1,W+,95,0,DiscreteOutp2_1
```

Importing this CSV, reflecting network variables in two different LonWorks nodes, results in the following screen shot.

LonWorks-BACnet Device Manager Configuration Tool v2.06

Connected: Sync:

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

No action Execute 12 network variables parsed from nvlist.csv.

Insert NV Append NV Add Field Delete

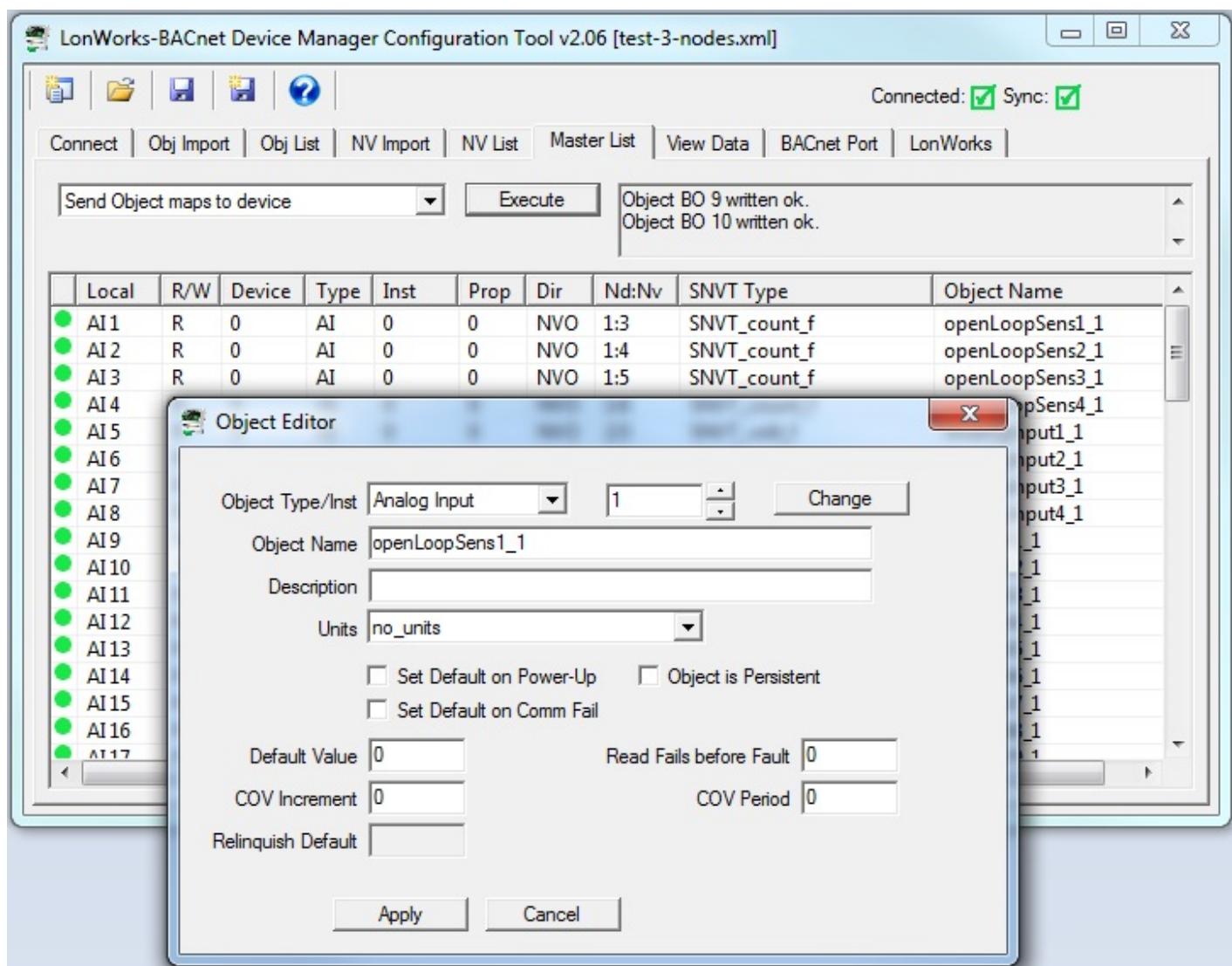
	Dir	Nd:Nv	Loc ...	SNVT Type	SNVT Category	NV Name
●	NVO	1:5	---	SNVT_volt_f	---	AnalogInput1_1
●	NVO	1:6	---	SNVT_volt_f	---	AnalogInput2_1
●	NVO	1:7	---	SNVT_volt_f	---	AnalogInput3_1
●	NVO	1:8	---	SNVT_volt_f	---	AnalogInput4_1
●	NVI	1:9	---	SNVT_switch	---	DiscreteOutp1_1
●	NVI	1:9	---	:: value	nvt_cat_unsigned_short	DiscreteOutp1_1_1
●	NVI	1:9	---	:: state	nvt_cat_signed_short	DiscreteOutp1_1_2
●	NVI	1:10	---	SNVT_switch	---	DiscreteOutp2_1
●	NVI	1:10	---	:: value	nvt_cat_unsigned_short	DiscreteOutp2_1_1
●	NVI	1:10	---	:: state	nvt_cat_signed_short	DiscreteOutp2_1_2
●	NVO	2:5	---	SNVT_volt_f	---	AnalogInput1_1
●	NVO	2:6	---	SNVT_volt_f	---	AnalogInput2_1
●	NVO	2:7	---	SNVT_volt_f	---	AnalogInput3_1
●	NVO	2:8	---	SNVT_volt_f	---	AnalogInput4_1
●	NVI	2:9	---	SNVT_switch	---	DiscreteOutp1_1
●	NVI	2:9	---	:: value	nvt_cat_unsigned_short	DiscreteOutp1_1_1
●	NVI	2:9	---	:: state	nvt_cat_signed_short	DiscreteOutp1_1_2



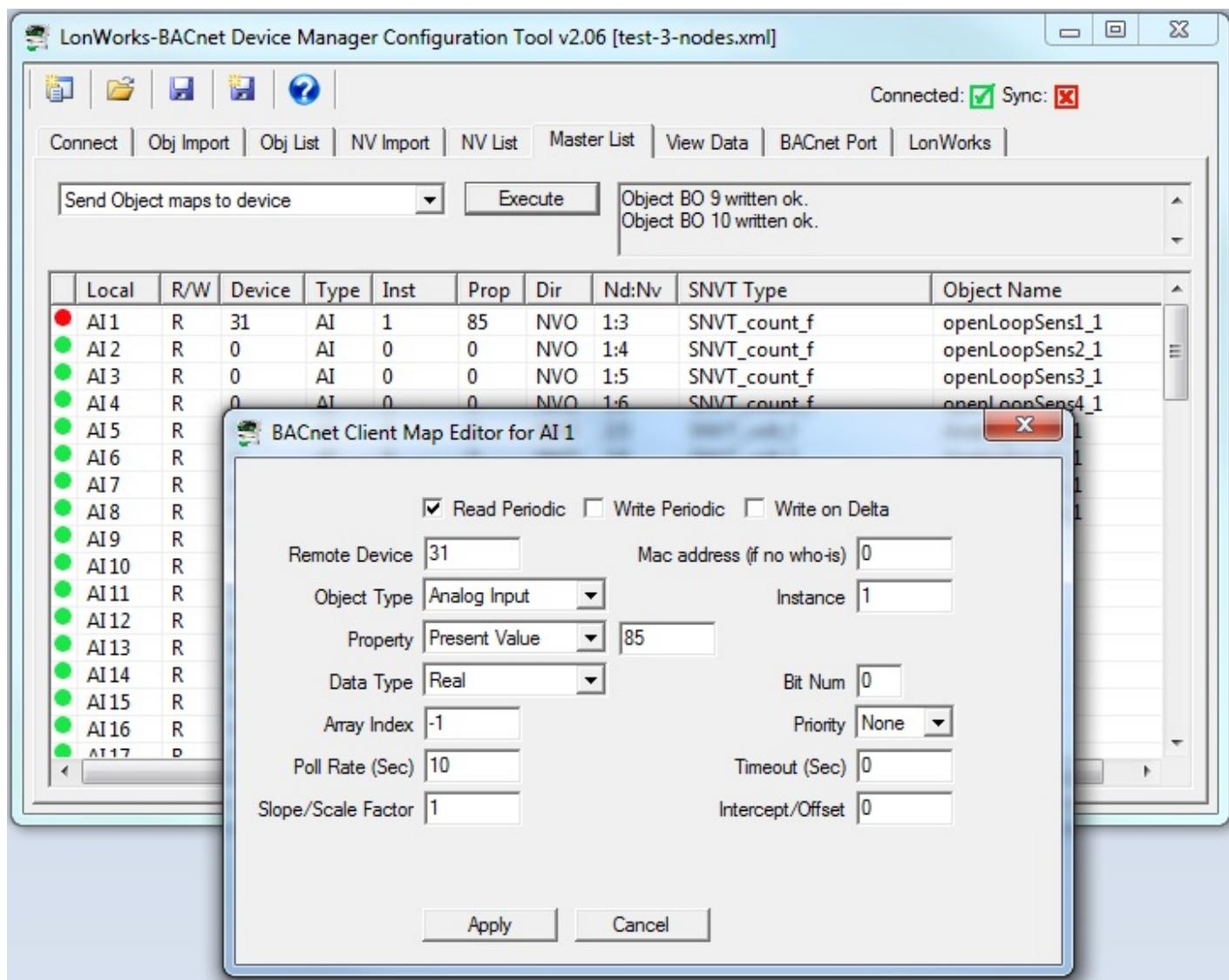
9 Tool 'Master List' Page

9.1 Editing Configuration from Master List Page

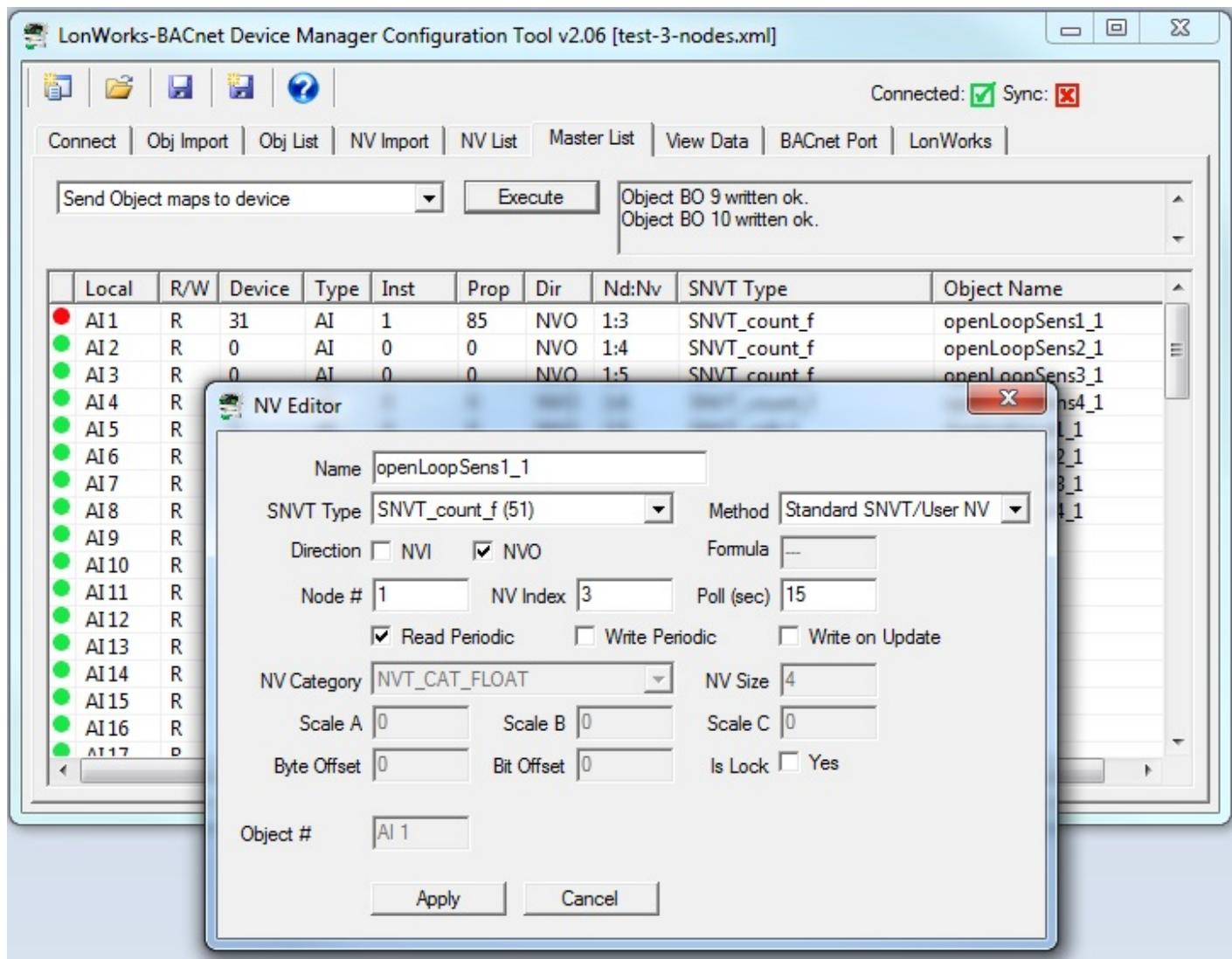
Clicking on the Object number column will open the Object Editor dialog as illustrated below. This is the same editor described in Section 6 of this user guide. Refer to that section for details about the BACnet Object Editor.



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.



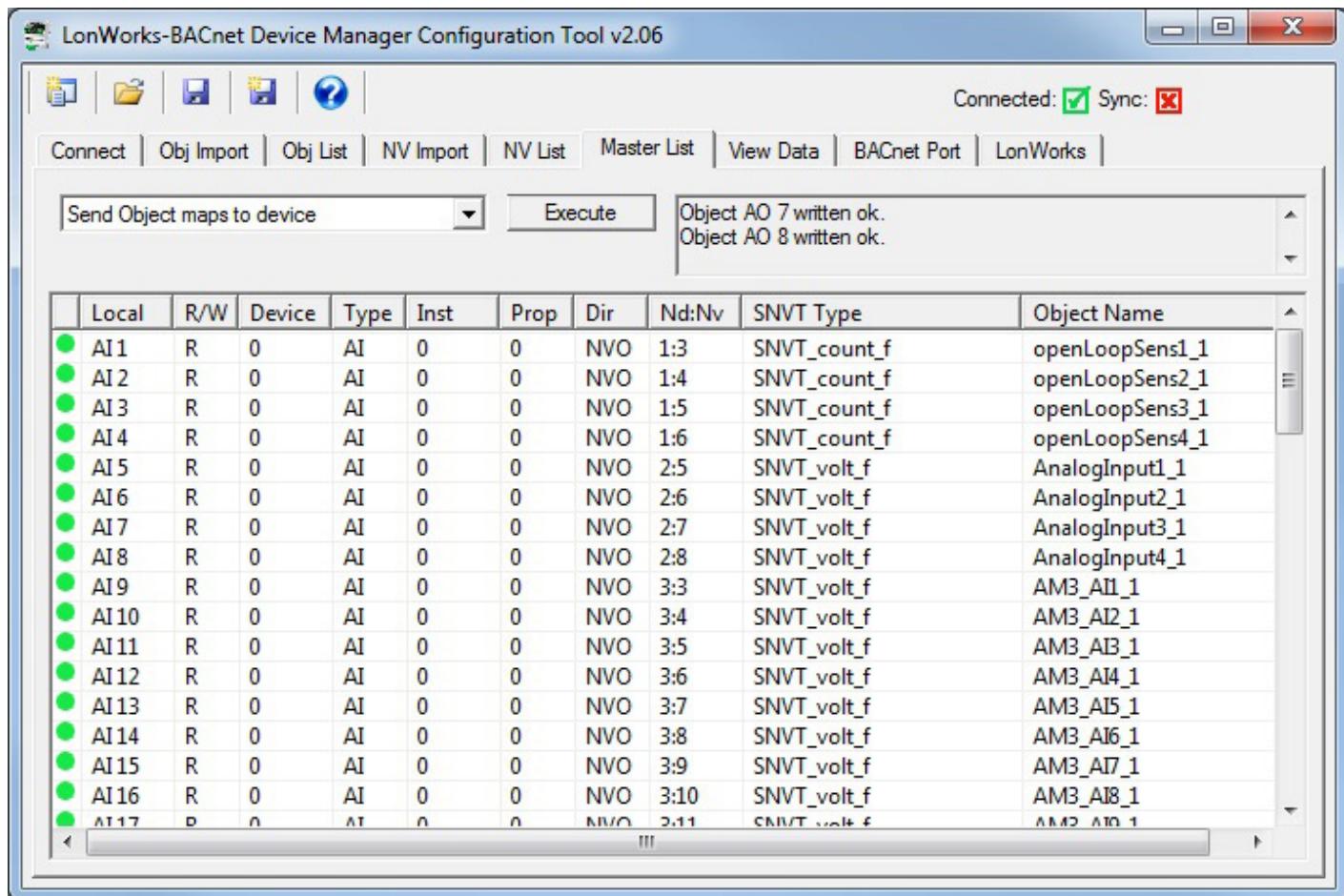
Clicking on the Network Variable area of a line on the Master List will open the NV Editor dialog as illustrated below. This is the same editor described in Section 8 of this user guide. Refer to that section for details about the NV Editor.



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, and generally in this order:

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



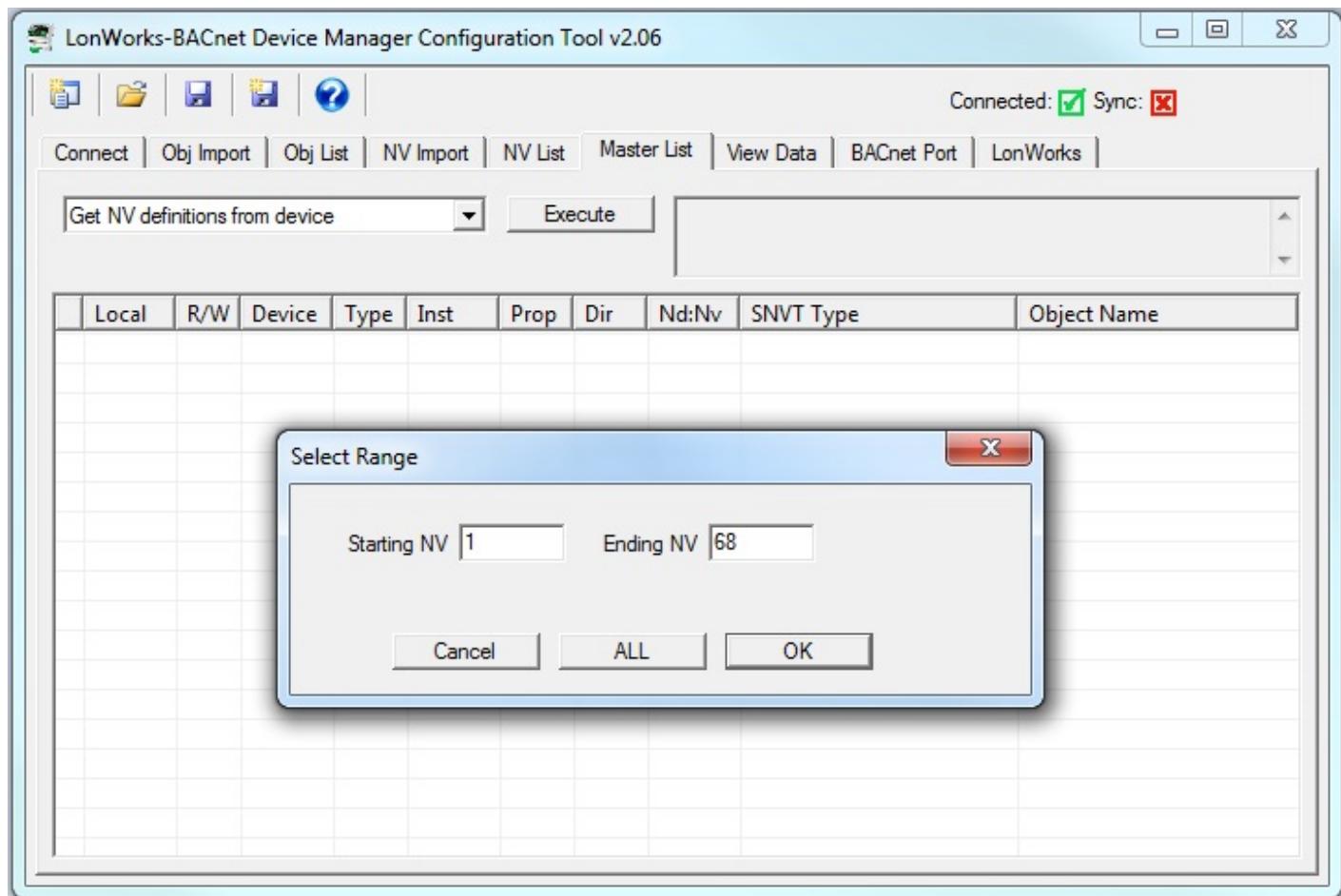
9.3 Getting Configuration From Device

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.

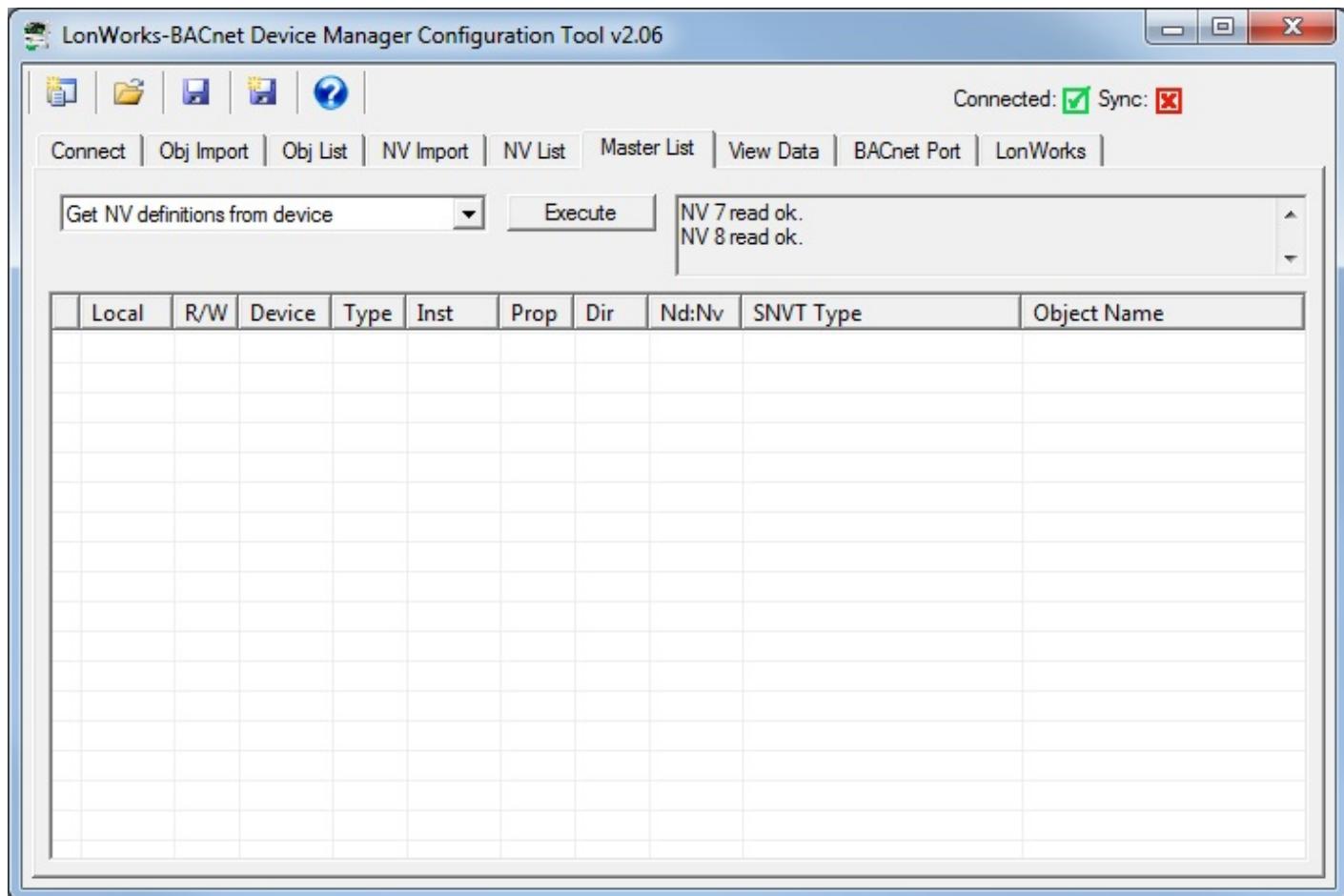
There are a total of four pages containing information that must be read from the gateway to obtain the complete configuration.

- 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 query 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.



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.

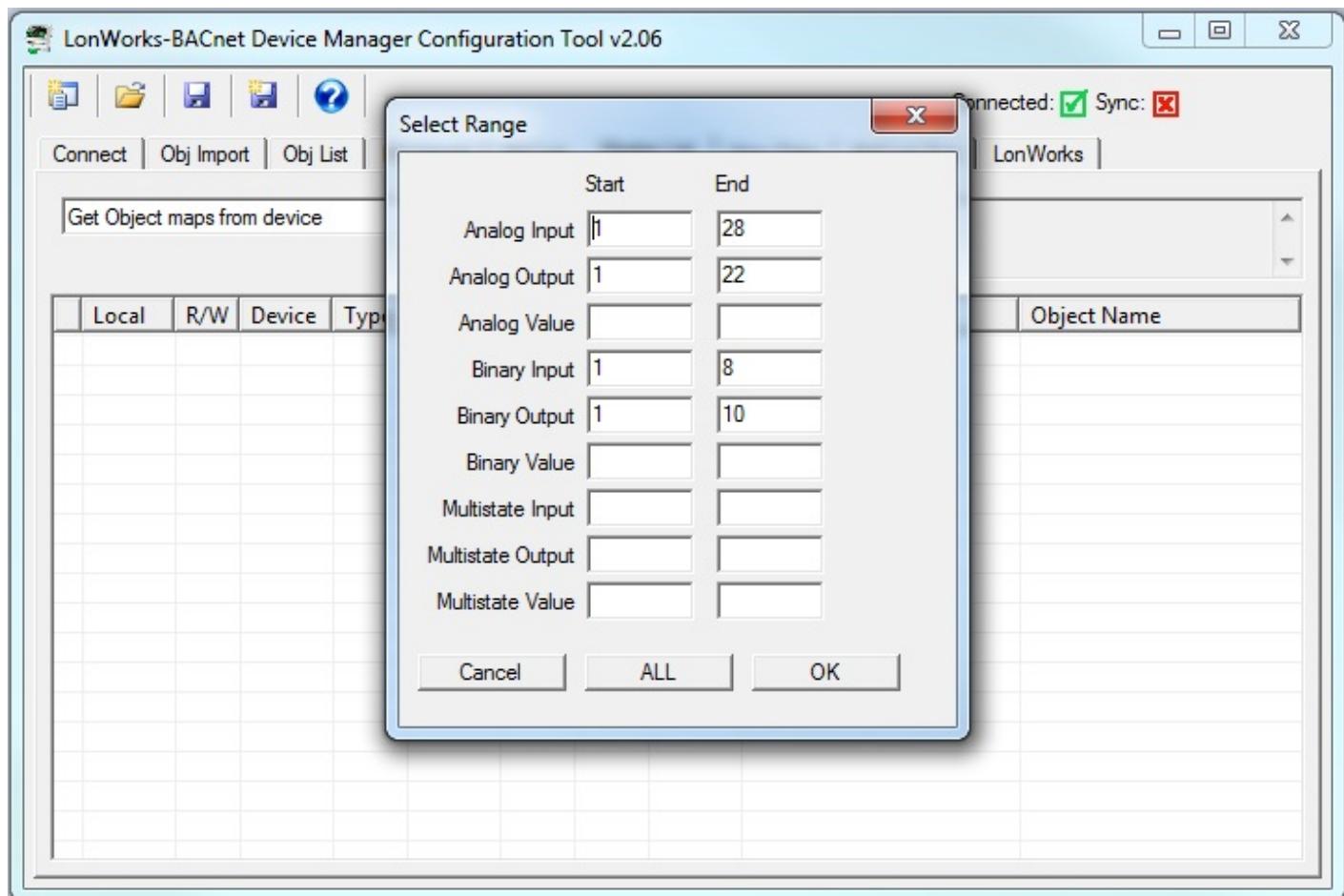


The NV List will appear something like the screen below following the Get NVs. Note that no local objects are yet known.

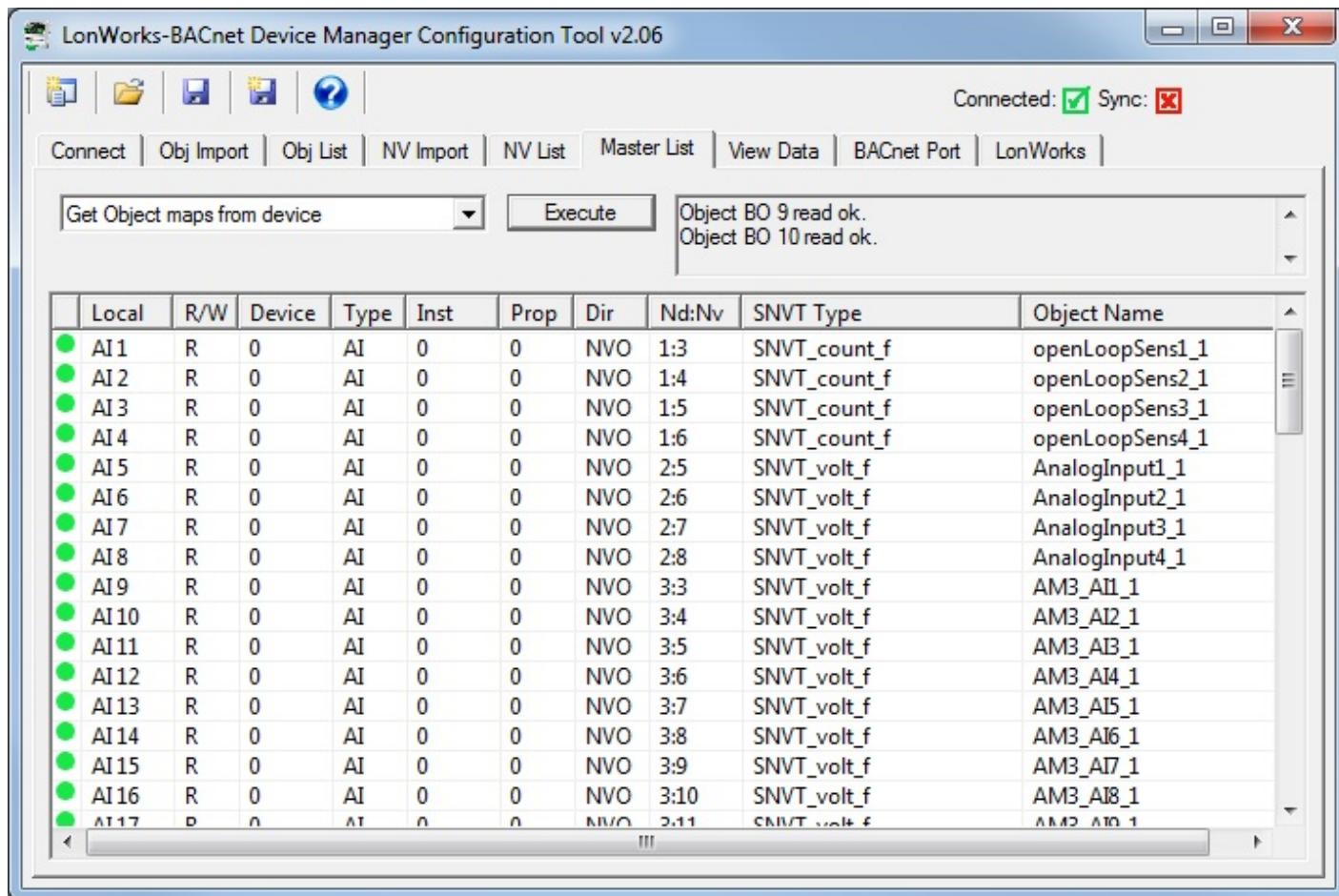
The screenshot shows the LonWorks-BACnet Device Manager Configuration Tool v2.06. The window title is "LonWorks-BACnet Device Manager Configuration Tool v2.06". The menu bar includes "File", "Edit", "View", "Tools", "Help", and "About". The toolbar contains icons for Connect, Obj Import, Obj List, NV Import, NV List, Master List, View Data, BACnet Port, and LonWorks. The status bar shows "Connected: Sync: ". The main window has tabs: "NV List" (selected), "Master List", "View Data", "BACnet Port", and "LonWorks". Below the tabs is a message box: "Send NV definitions to device" dropdown, "Execute" button, and a log area: "NV 1 read ok.", "NV 2 read ok.", "NV 3 read ok.". Buttons below the log are "Insert NV", "Append NV", "Add Field", and "Delete". The main table has columns: Dir, Nd:Nv, Loc ..., SNVT Type, SNVT Category, and NV Name. The table lists NV definitions:

Dir	Nd:Nv	Loc ...	SNVT Type	SNVT Category	NV Name
● NVO	1:3	---	SNVT_count_f	---	openLoopSens1_1
● NVO	1:4	---	SNVT_count_f	---	openLoopSens2_1
● NVO	1:5	---	SNVT_count_f	---	openLoopSens3_1
● NVO	1:6	---	SNVT_count_f	---	openLoopSens4_1
● NVO	1:7	---	SNVT_switch	---	openLoopSens5_1
● NVO	1:7	---	:: value	nvt_cat_unsigned_short	openLoopSens5_1_1
● NVO	1:7	---	:: state	nvt_cat_signed_short	openLoopSens5_1_2
● NVO	1:8	---	SNVT_switch	---	openLoopSens6_1
● NVO	1:8	---	:: value	nvt_cat_unsigned_short	openLoopSens6_1_1
● NVO	1:8	---	:: state	nvt_cat_signed_short	openLoopSens6_1_2
● NVO	1:9	---	SNVT_switch	---	openLoopSens7_1
● NVO	1:9	---	:: value	nvt_cat_unsigned_short	openLoopSens7_1_1
● NVO	1:9	---	:: state	nvt_cat_signed_short	openLoopSens7_1_2
● NVO	1:10	---	SNVT_switch	---	openLoopSens8_1
● NVO	1:10	---	:: value	nvt_cat_unsigned_short	openLoopSens8_1_1
● NVO	1:10	---	:: state	nvt_cat_signed_short	openLoopSens8_1_2
● NVO	1:11	---	CNV/T switch	---	openLoopSens0_1

Upon clicking Execute with "Get Object maps from device" selected, the tool will query 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.



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

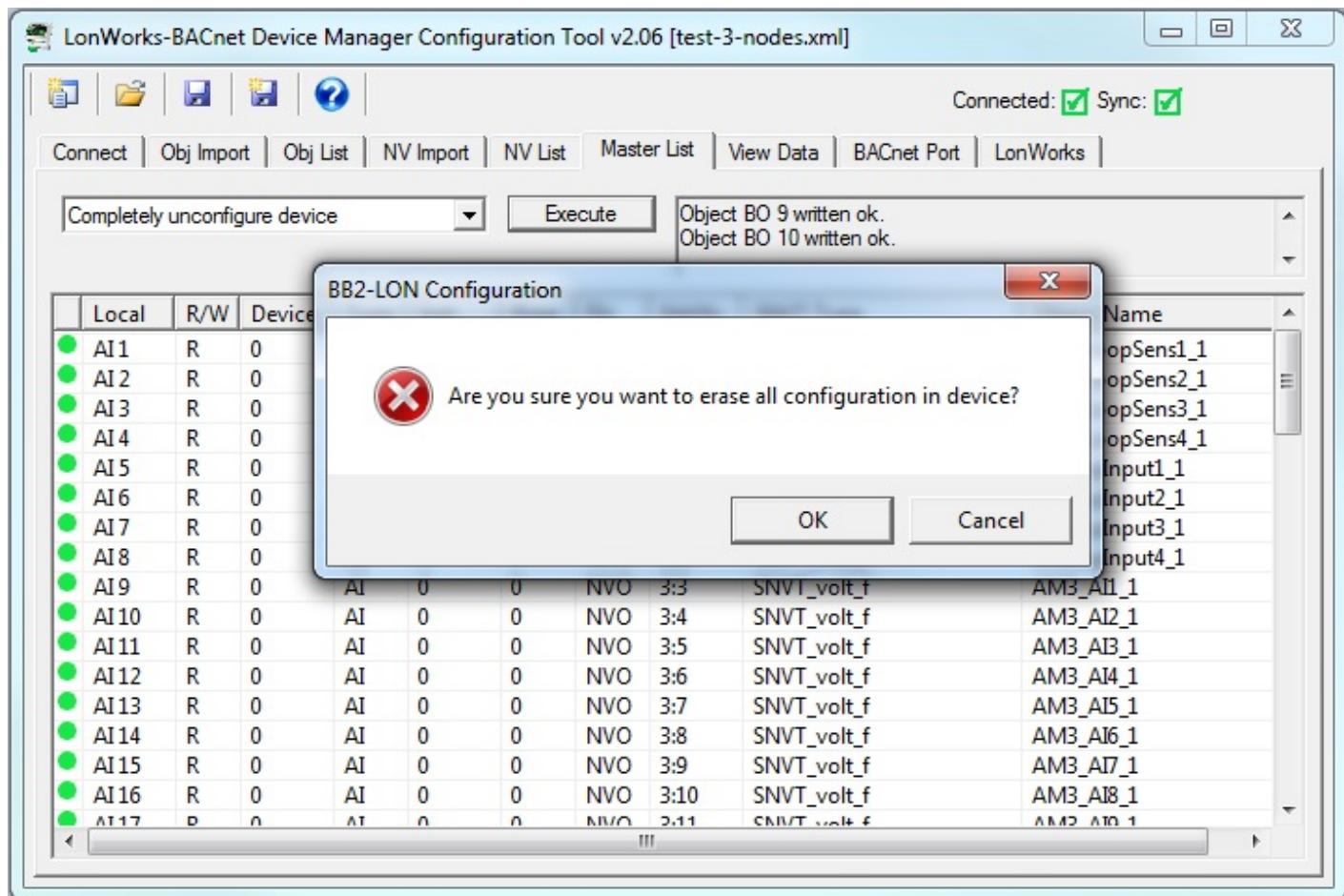


9.4 Fixing Conflicts

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 60 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.



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. Note that the NV List and Object List will be cleared in the device but not in the configuration tool. You will need to resend the NV List, Object List, and Node List (LonWorks page) to become fully operational.

If you want to also clear the tool, click the 'new' icon in the upper left corner of the toolbar.

LonWorks-BACnet Device Manager Configuration Tool v2.06 [test-3-nodes.xml]

Connected: Sync:

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

Completely unconfigure device Waiting... (takes up to 60 seconds).
Device cleared.

	Local	R/W	Device	Type	Inst	Prop	Dir	Nd:Nv	SNVT Type	Object Name
●	AI1	R	0	AI	0	0	NVO	1:3	SNVT_count_f	openLoopSens1_1
●	AI2	R	0	AI	0	0	NVO	1:4	SNVT_count_f	openLoopSens2_1
●	AI3	R	0	AI	0	0	NVO	1:5	SNVT_count_f	openLoopSens3_1
●	AI4	R	0	AI	0	0	NVO	1:6	SNVT_count_f	openLoopSens4_1
●	AI5	R	0	AI	0	0	NVO	2:5	SNVT_volt_f	AnalogInput1_1
●	AI6	R	0	AI	0	0	NVO	2:6	SNVT_volt_f	AnalogInput2_1
●	AI7	R	0	AI	0	0	NVO	2:7	SNVT_volt_f	AnalogInput3_1
●	AI8	R	0	AI	0	0	NVO	2:8	SNVT_volt_f	AnalogInput4_1
●	AI9	R	0	AI	0	0	NVO	3:3	SNVT_volt_f	AM3_AI1_1
●	AI10	R	0	AI	0	0	NVO	3:4	SNVT_volt_f	AM3_AI2_1
●	AI11	R	0	AI	0	0	NVO	3:5	SNVT_volt_f	AM3_AI3_1
●	AI12	R	0	AI	0	0	NVO	3:6	SNVT_volt_f	AM3_AI4_1
●	AI13	R	0	AI	0	0	NVO	3:7	SNVT_volt_f	AM3_AI5_1
●	AI14	R	0	AI	0	0	NVO	3:8	SNVT_volt_f	AM3_AI6_1
●	AI15	R	0	AI	0	0	NVO	3:9	SNVT_volt_f	AM3_AI7_1
●	AI16	R	0	AI	0	0	NVO	3:10	SNVT_volt_f	AM3_AI8_1
●	AT17	R	0	AT	0	0	NVO	3:11	SNVT_volt_f	AM3_AI9_1



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.

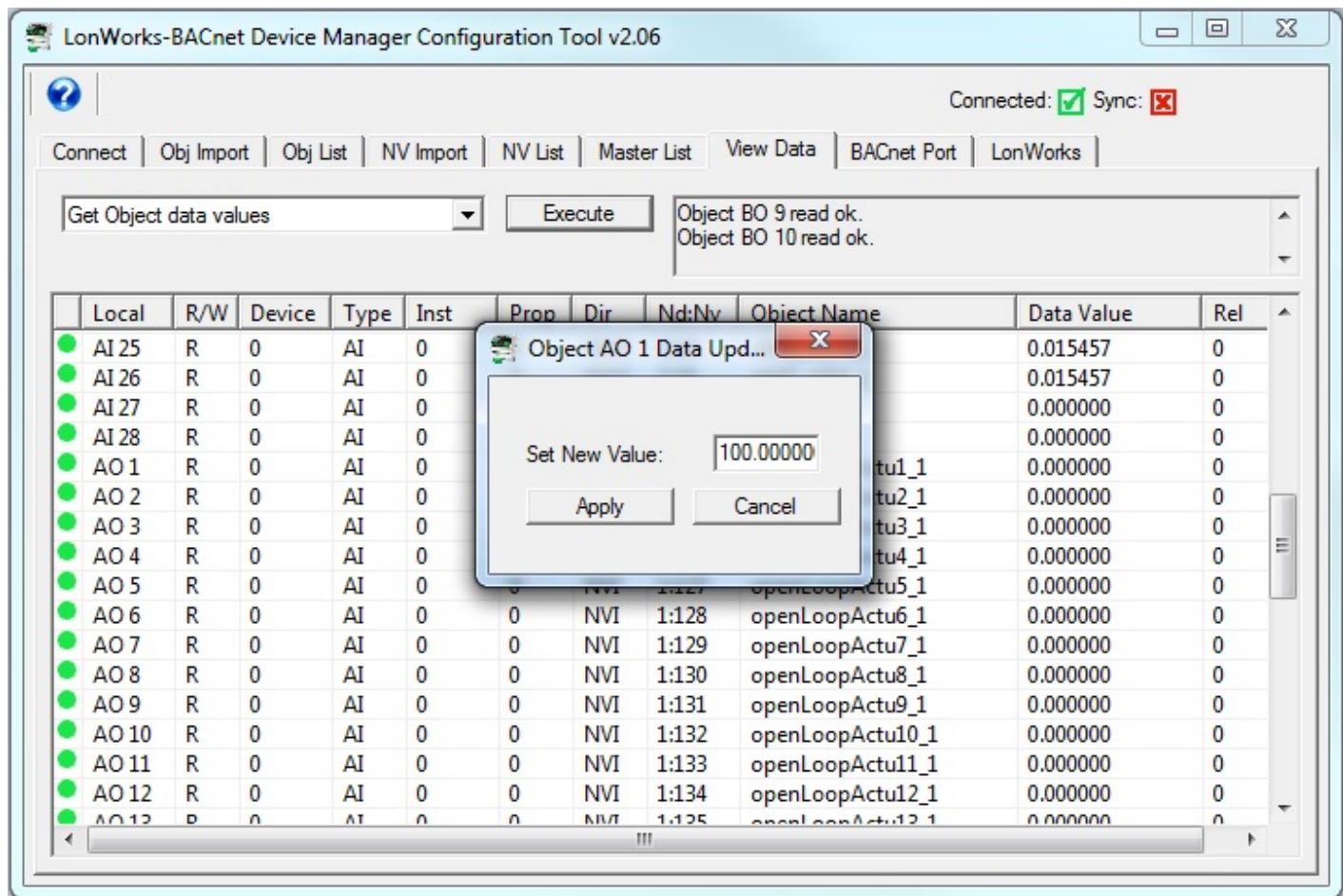
The most user friendly way to view data from the LonWorks side is to use the network management tool's NV browser, as found in Echelon's LonMaker. You can also use Nodeutil to view NV data in raw form via the LonWorks network. The third option is that you can view NV data in raw form via the USB diagnostic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

Local	R/W	Device	Type	Inst	Prop	Dir	Nd:Nv	Object Name	Data Value	Rel
AI1	R	0	AI	0	0	NVO	1:3	openLoopSens1_1	19.000000	0
AI2	R	0	AI	0	0	NVO	1:4	openLoopSens2_1	20.000000	0
AI3	R	0	AI	0	0	NVO	1:5	openLoopSens3_1	21.000000	0
AI4	R	0	AI	0	0	NVO	1:6	openLoopSens4_1	22.000000	0
AI5	R	0	AI	0	0	NVO	2:5	AnalogInput1_1	0.000000	0
AI6	R	0	AI	0	0	NVO	2:6	AnalogInput2_1	12.329000	0
AI7	R	0	AI	0	0	NVO	2:7	AnalogInput3_1	0.000000	0
AI8	R	0	AI	0	0	NVO	2:8	AnalogInput4_1	0.000000	0
AI9	R	0	AI	0	0	NVO	3:3	AM3_AI1_1	0.007917	0
AI10	R	0	AI	0	0	NVO	3:4	AM3_AI2_1	10.000000	0
AI11	R	0	AI	0	0	NVO	3:5	AM3_AI3_1	0.012441	0
AI12	R	0	AI	0	0	NVO	3:6	AM3_AI4_1	0.012441	0
AI13	R	0	AI	0	0	NVO	3:7	AM3_AI5_1	0.012818	0
AI14	R	0	AI	0	0	NVO	3:8	AM3_AI6_1	0.013195	0
AI15	R	0	AI	0	0	NVO	3:9	AM3_AI7_1	0.013572	0
AI16	R	0	AI	0	0	NVO	3:10	AM3_AI8_1	0.013949	0
AI17	R	0	AI	0	0	NVO	3:11	AM3_AI9_1	0.011697	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.

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.



10.3 Diagnostics - Object Reliability

The reliability code will provide an indication of problems if non-zero. The following reliability codes can occur when the Babel Buster is operating as a BACnet client (polling other BACnet devices) and there is a problem related to the attempt to access the remote BACnet device:

- 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.

The following reliability codes pertain to the LonWorks side, and indicate a problem attempting to communicate with the LonWorks device:

90 - NV Fail - most often incorrect NV index or incorrect NV size if writing to the LonWorks device

91 - Node Fail - unable to communicate with the node

92 - API Error - indicates a problem with the host processor's attempt to communicate with the internal LonWorks processor.

Note that "Node Fail" can have a multitude of causes. If a node had previously communicated successfully and then fails, it can mean something simple like disconnected network cable. However, if the node has never communicated, there is probably a fundamental configuration problem and checking node status on the LonWorks page will indicate that the node is not in the "Ready" state. Proceed to attempt to commission the device as discussed in the LonWorks section.

The following screen shot is what you would see when node #2 is simply disconnected. If you were using an external BACnet client to read BACnet objects from this Babel Buster, Analog Inputs 5 through 8 would be reporting a fault in the status property, and reliability code 91.

The screenshot shows the LonWorks-BACnet Device Manager Configuration Tool v2.06. The window title is "LonWorks-BACnet Device Manager Configuration Tool v2.06". At the top right, there are buttons for "Connected" (checked) and "Sync" (unchecked). Below the title bar is a menu bar with tabs: Connect, Obj Import, Obj List, NV Import, NV List, Master List, View Data (selected), BACnet Port, and LonWorks. A toolbar below the menu bar includes a question mark icon, a dropdown menu, and a "Execute" button. The main area contains a command history window with the text "Object BO 9 read ok." and "Object BO 10 read ok.". Below this is a table of objects:

	AI1	R	0	AI	0	0	NVO	1:3	openLoopSens1_1	19.000000	0
●	AI2	R	0	AI	0	0	NVO	1:4	openLoopSens2_1	20.000000	0
●	AI3	R	0	AI	0	0	NVO	1:5	openLoopSens3_1	21.000000	0
●	AI4	R	0	AI	0	0	NVO	1:6	openLoopSens4_1	22.000000	0
●	AI5	R	0	AI	0	0	NVO	2:5	AnalogInput1_1	0.000000	91
●	AI6	R	0	AI	0	0	NVO	2:6	AnalogInput2_1	12.329000	91
●	AI7	R	0	AI	0	0	NVO	2:7	AnalogInput3_1	0.000000	91
●	AI8	R	0	AI	0	0	NVO	2:8	AnalogInput4_1	0.000000	91
●	AI9	R	0	AI	0	0	NVO	3:3	AM3_AI1_1	0.007917	0
●	AI10	R	0	AI	0	0	NVO	3:4	AM3_AI2_1	10.000000	0
●	AI11	R	0	AI	0	0	NVO	3:5	AM3_AI3_1	0.012441	0
●	AI12	R	0	AI	0	0	NVO	3:6	AM3_AI4_1	0.012441	0
●	AI13	R	0	AI	0	0	NVO	3:7	AM3_AI5_1	0.012818	0
●	AI14	R	0	AI	0	0	NVO	3:8	AM3_AI6_1	0.013195	0
●	AI15	R	0	AI	0	0	NVO	3:9	AM3_AI7_1	0.013572	0
●	AI16	R	0	AI	0	0	NVO	3:10	AM3_AI8_1	0.013949	0
●	AT17	D	0	AT	0	0	NVO	3:11	AM3_AT1_1	0.011697	0



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 Device Manager Configuration Tool v2.06

Connected: Sync:

Object ID (device instance)

Object Allocation	Present	Pending
Basic Objects	211	
Commandable Objects	45	
Analog Input Count	57	57
Analog Output Count	35	35
Analog Value Count	0	0
Binary Input Count	8	8
Binary Output Count	10	10
Binary Value Count	0	0
Multistate Input Count	0	0
Multistate Output Count	0	0
Multistate Value Count	0	0

Device Name

Device Description

Device Location

Max Master

MS/TP Baud Rate ▾

MS/TP Station/MAC Addr

Password

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-NB and BB2-7020-NB 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-NB 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 will have serious communication problems.

11.4 IP Port Settings

The BACnet IP port settings will appear if BB2-7020-NB 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.

LonWorks-BACnet Device Manager Configuration Tool v2.06

Connected: Sync:

Object ID (device instance)

Device Name

Device Description

Device Location

IP address, port

Subnet mask

Network gateway

Password

Object Allocation Present Pending

Basic Objects	211
Commandable Objects	45
Analog Input Count	57
Analog Output Count	35
Analog Value Count	0
Binary Input Count	8
Binary Output Count	10
Binary Value Count	0
Multistate Input Count	0
Multistate Output Count	0
Multistate Value Count	0

Device configuration read from device.



12 Tool 'LonWorks' Page

12.1 Viewing LonWorks Identity of the Gateway

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 90:00:17:47:1E:04:04:01 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-NB or BB2-7020-NB.

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.

 A screenshot of the LonWorks-BACnet Device Manager Configuration Tool v2.06. The window title is "LonWorks-BACnet Device Manager Configuration Tool v2.06". At the top, there is a toolbar with icons for Help, Connect, Obj Import, Obj List, NV Import, NV List, Master List, View Data, BACnet Port, and LonWorks. The "Connected" status is shown as "Connected: [checkmark] Sync: [checkbox]".

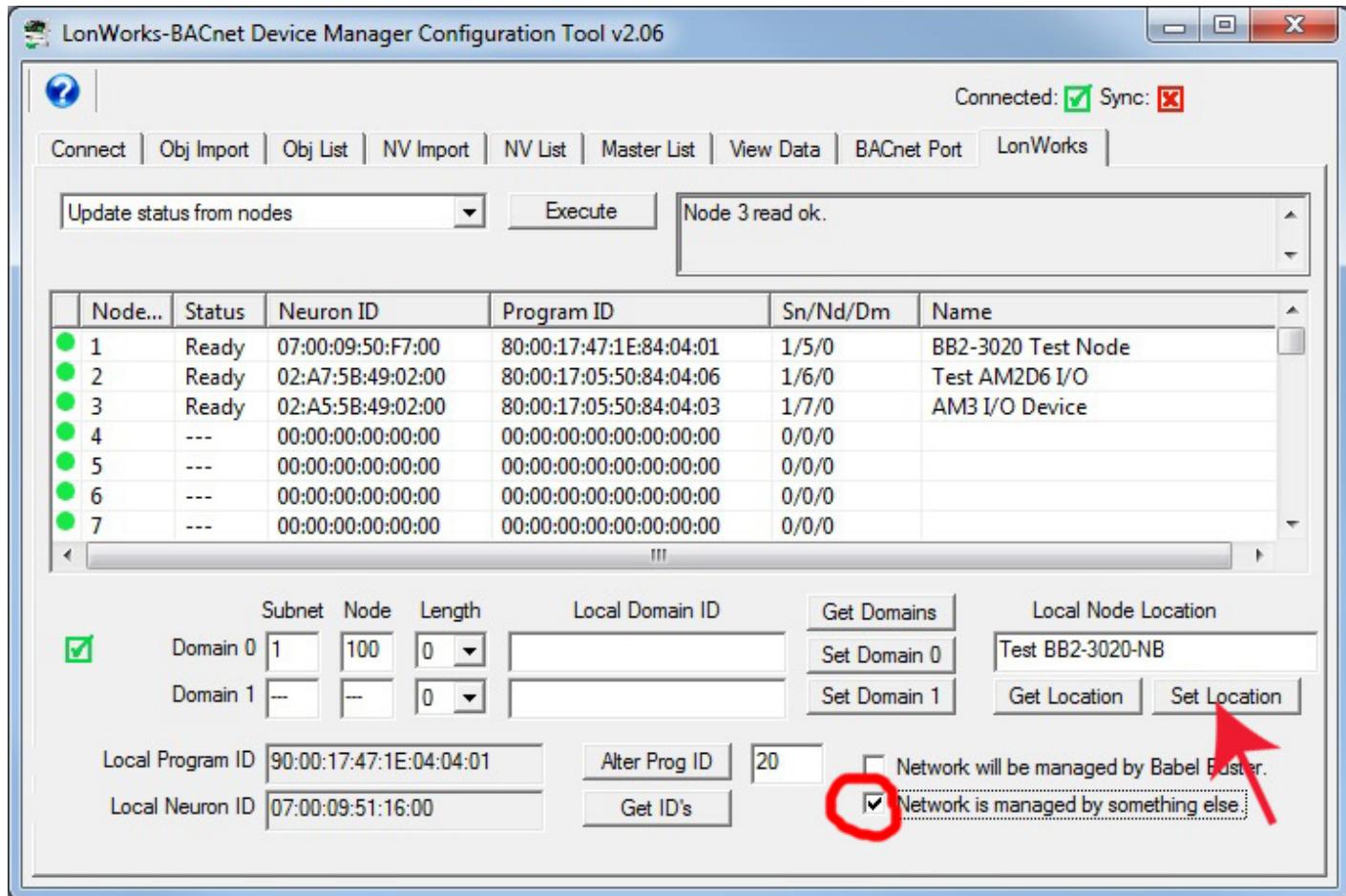
The main interface has a message box saying "Node 3 read ok." Below it is a table with columns: Node..., Status, Neuron ID, Program ID, Sn/Nd/Dm, and Name. The table lists nodes 1 through 7, all marked as Ready. The "Program ID" column shows values like 80:00:17:47:1E:84:04:01 for node 1 and 00:00:00:00:00:00:00:00 for nodes 4-7. The "Name" column shows "BB2-3020 Test Node", "Test AM2D6 I/O", and "AM3 I/O Device" for nodes 1, 2, and 3 respectively.

At the bottom, there is a red-bordered configuration panel with fields for Subnet (1), Node (100), Length (0), Local Domain ID, and Local Node Location (Test BB2-3020-NB). It also contains checkboxes for "Get Domains", "Set Domain 0", "Set Domain 1", "Get Location", and "Set Location". Below this are fields for Local Program ID (90:00:17:47:1E:04:04:01), Alter Prog ID (20), Local Neuron ID (07:00:09:51:16:00), and Get ID's. There are also checkboxes for "Network will be managed by Babel Buster" and "Network is managed by something else".

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, if the gateway is being installed on a network where 'Network is managed by something else'.

You have the option of viewing current domain settings using the Get Domains button, or setting the domains with the respective Set Domain buttons. If the gateway is being used on a network where 'Network is managed by something else', you will need to find out via the network management tool involved what the domain should be, and set it accordingly. If 'Network will be managed by Babel Buster', then the default should work, unless you run into trouble as noted in Appendix B.3 or B.4.

IMPORTANT: If you change the state of "Network is/will be managed..." as illustrated below, you MUST click 'Set Location' to send this change of state to the Babel Buster gateway device. While you may not care about the location string, the state of network management is included in the packet of information sent with location.



12.2 Changing Program ID of the Gateway

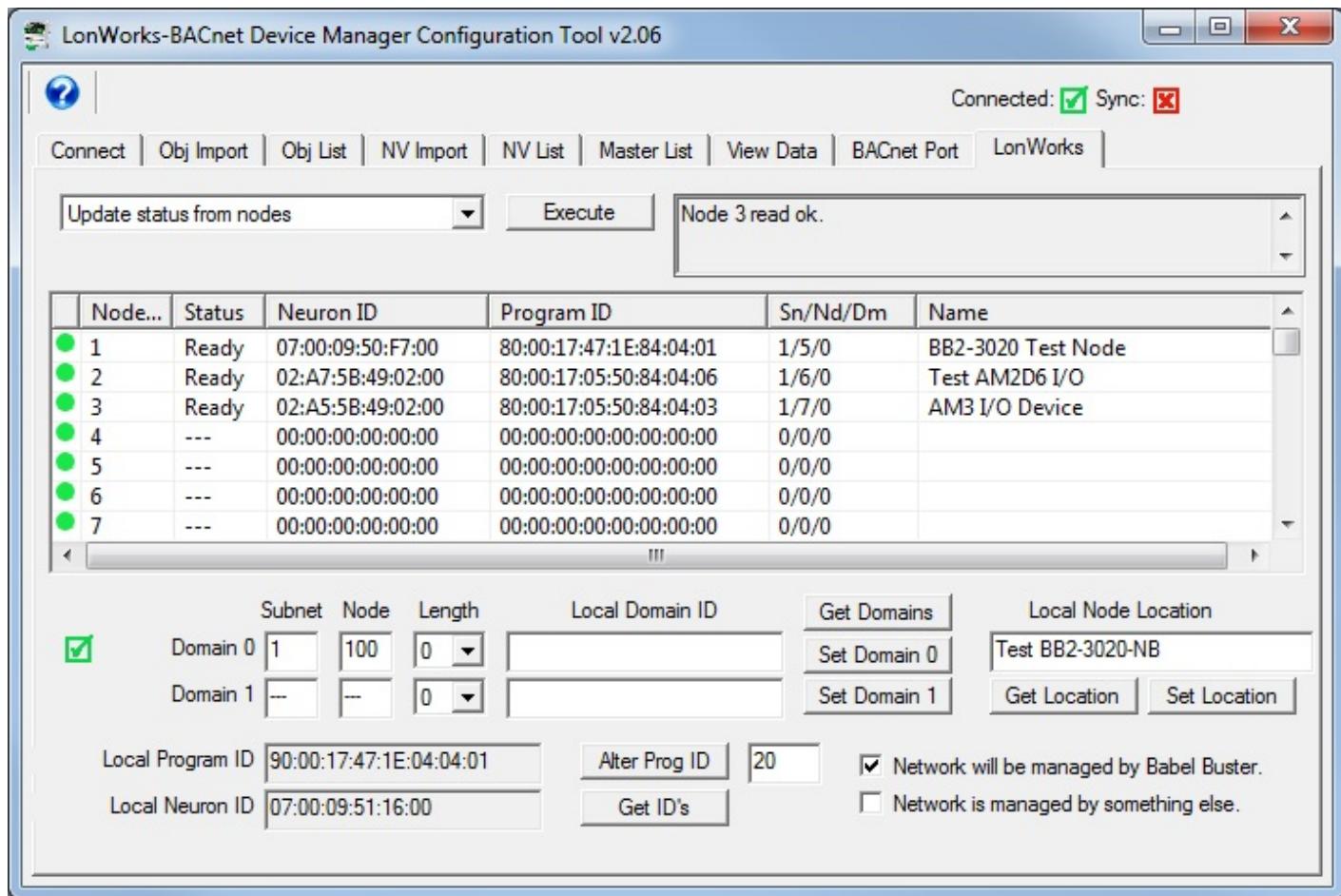
Changing program ID is only applicable if 'Network will be managed by something else' meaning Babel Buster is not in control of the network. You should normally have no reason to change the program ID. However, if you are pre-configuring multiple Babel Buster gateways and require them to have different program IDs, then you may use this option.

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.

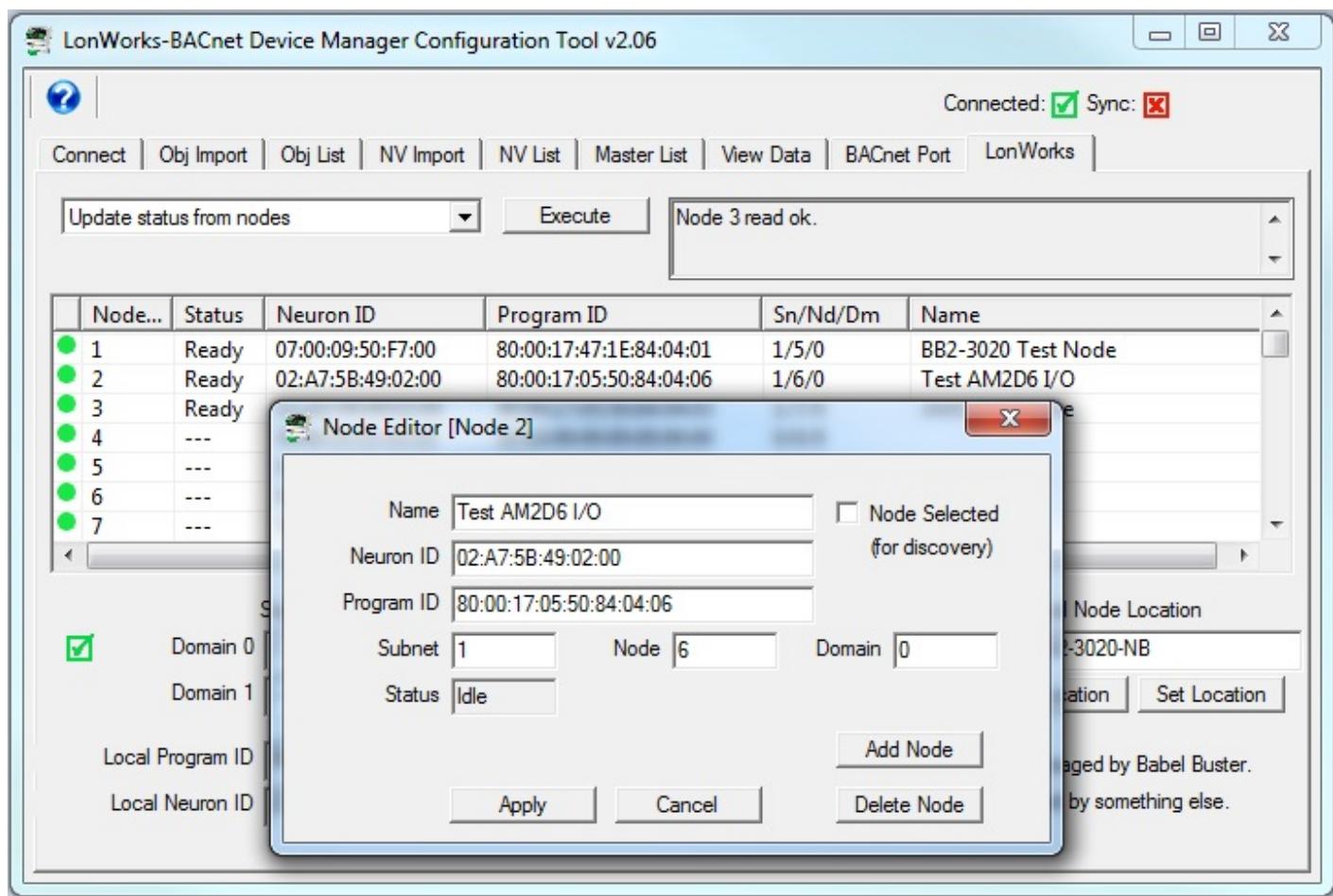
12.3 Viewing Identity and Status of Other LonWorks Devices

To view the identity and status of whatever LonWorks devices are known to the Babel Buster gateway, select 'Get node list from device', and click Execute. When it completes, then select 'Update status from

nodes' and click Execute again. You will now see the list of known LonWorks devices and a status indicating whether or not they are communicating. If they are communicating normally, the status will be listed as "Ready". If not, then a code indicating where the gateway is at in its attempt will be displayed instead.



There is more than one way to intially populate the node list. One way is to manually enter a Neuron ID and let the gateway try to talk to it. There are a couple of more automates ways of letting the gateway just to find out who's out there, as discussed in following sections. In any event, you can edit the information for a given node by double-clicking that node in the list to get the Node Editor dialog to pop up. If you make changes, enter a new Neuron ID, or discover new nodes as discussed further on, you will need to send that node configuration to the gateway before the gateway will begin attempting to communicate with it.



Node status will hopefully indicate "Ready". But it may indicate other codes showing the gateway's progress in attempting to communicate. The most common "non-ready" indication will be "SFN". This means it is completely unable to reach that particular node.

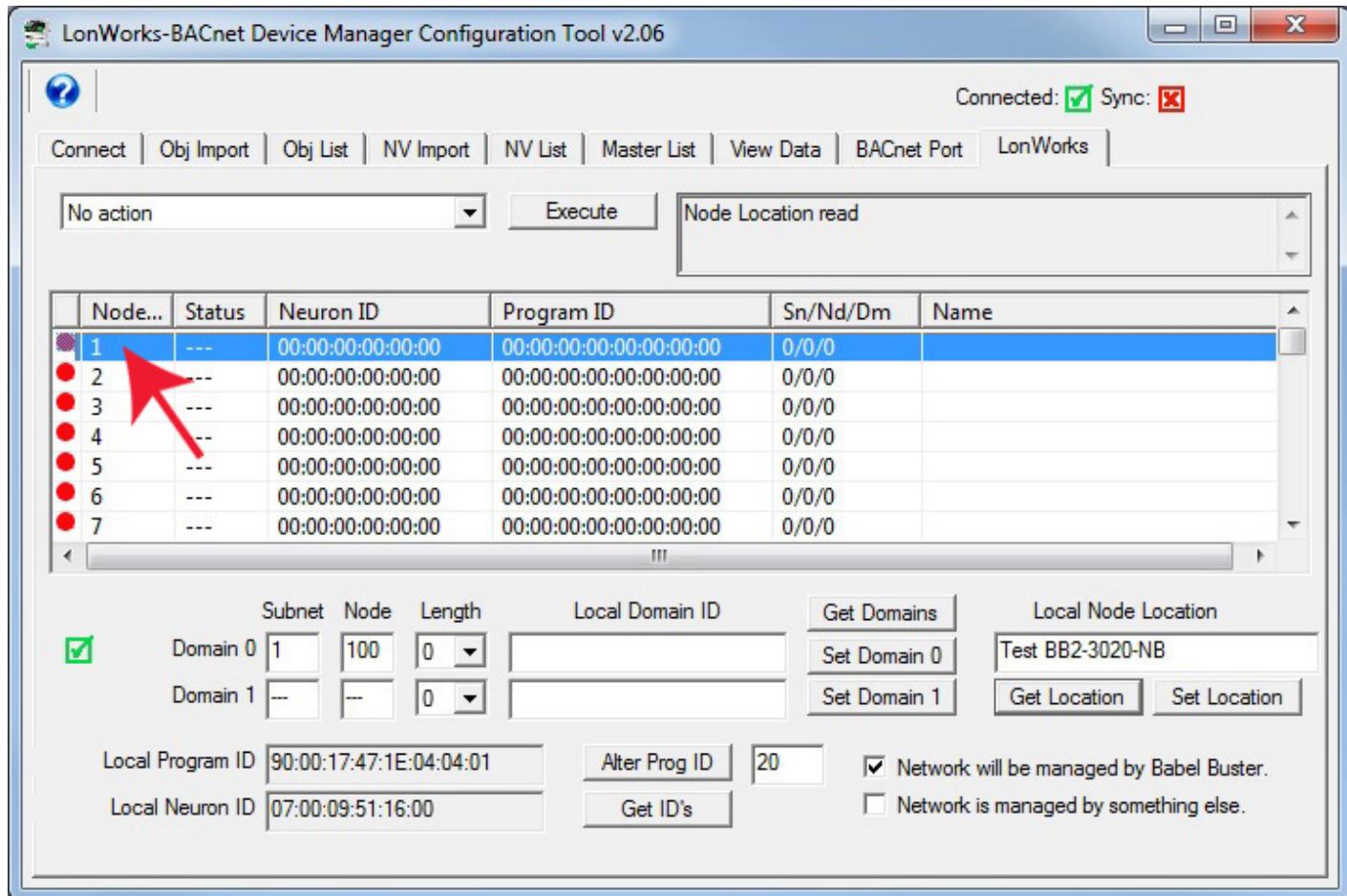
The meaning of the various letters that can potentially show up are as follows. Not all of these are error indications, some are merely progress indicators.

- S - Service pin, means this node table entry has a valid Neuron ID
- D - Domain set was successful
- Q - Domain query was successful
- R - Ready - will normally be replaced by "Ready" indication
- W - Domain query came back with wrong subnet/node
- F - Failed to set domain table in the LonWorks device at the Neuron ID shown
- N - Domain query failed (almost always implied by 'F')
- E - API error - there has been a miscommunication between the two processors in the gateway (contact tech support)

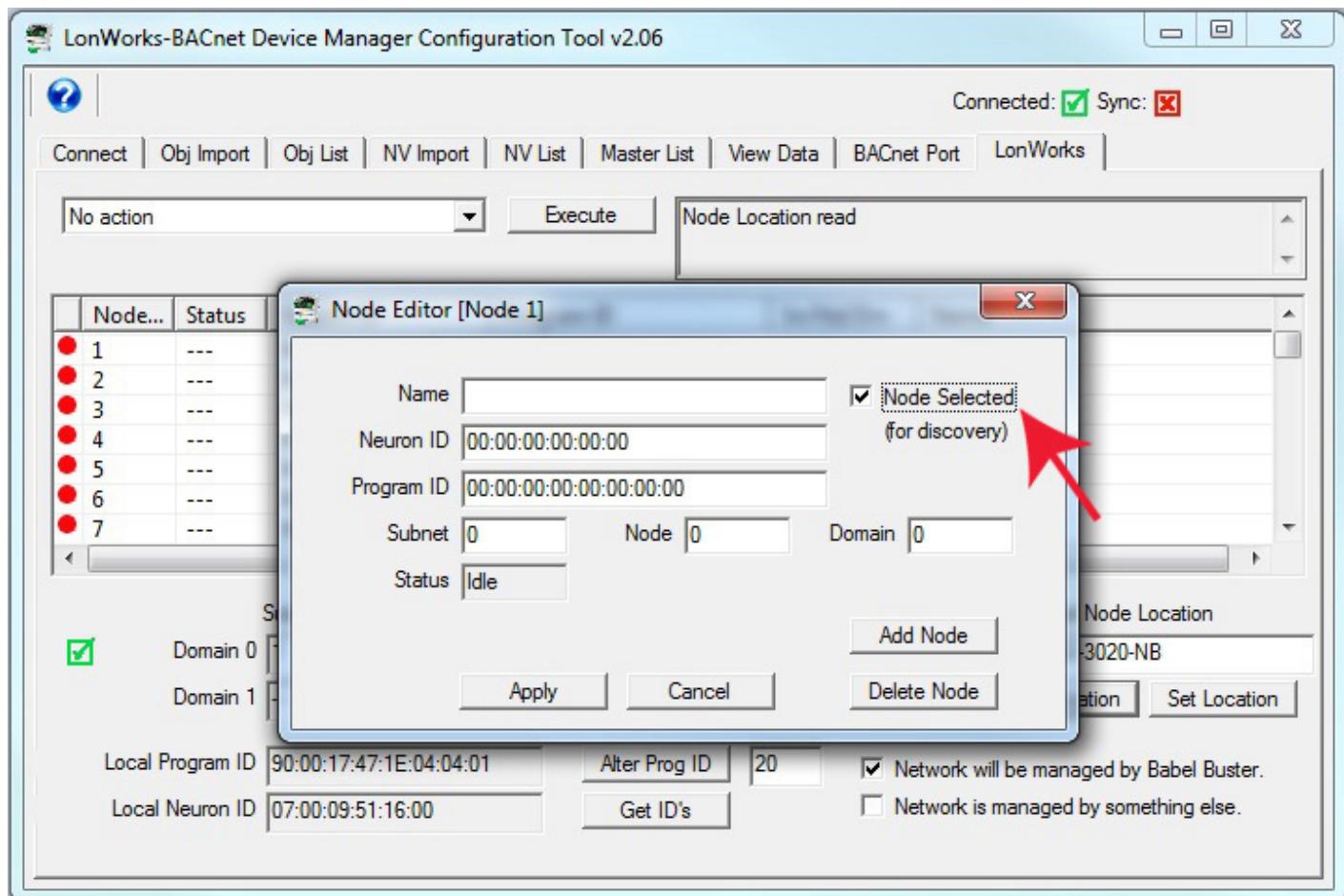
12.4 Node Discovery Using Service Pin

Virtually all LonWorks devices have a "Service Pin", or service button, also sometimes denoted as an 'Install' button. A node that cannot be discovered via network query due to the fact that its domain table is set to something unknown to (and incompatible with) the Babel Buster gateway can still be discovered using the service pin/button method.

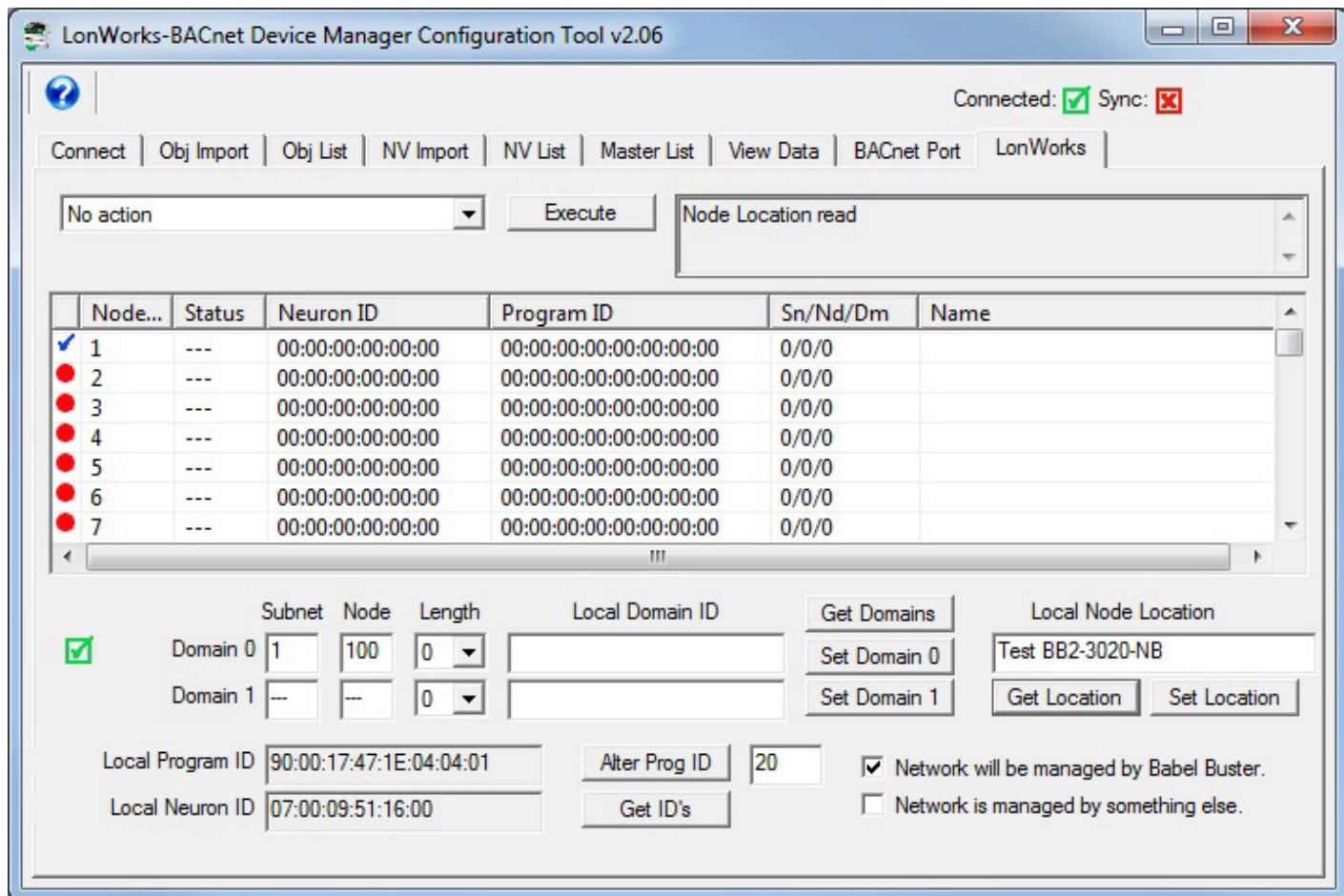
To discover a node using the service pin method, start by double clicking on an unused entry in the node table to open the Node Editor dialog.



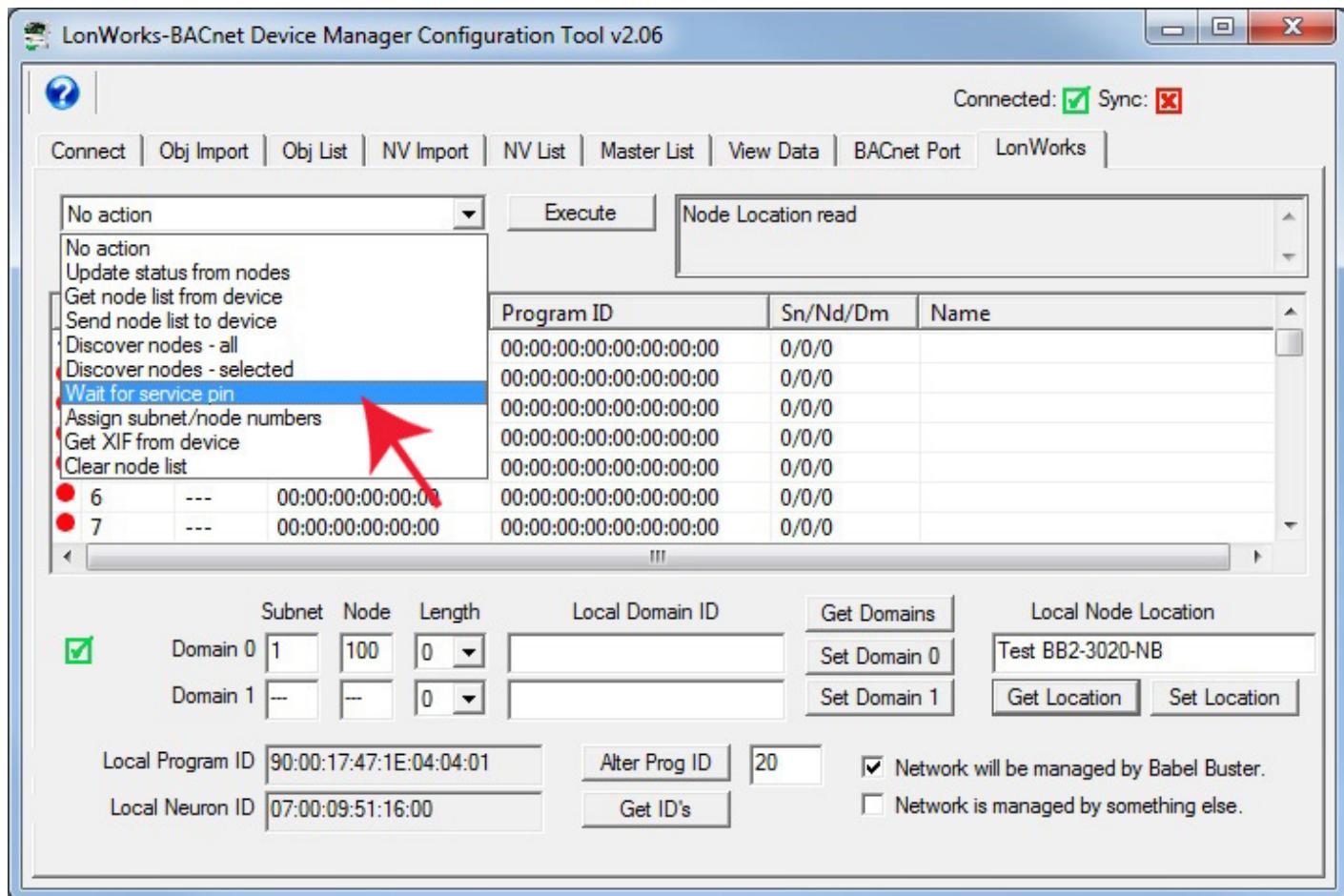
Click the 'Node Selected' box. This tells the tool that we are going to put whatever we find in this spot.



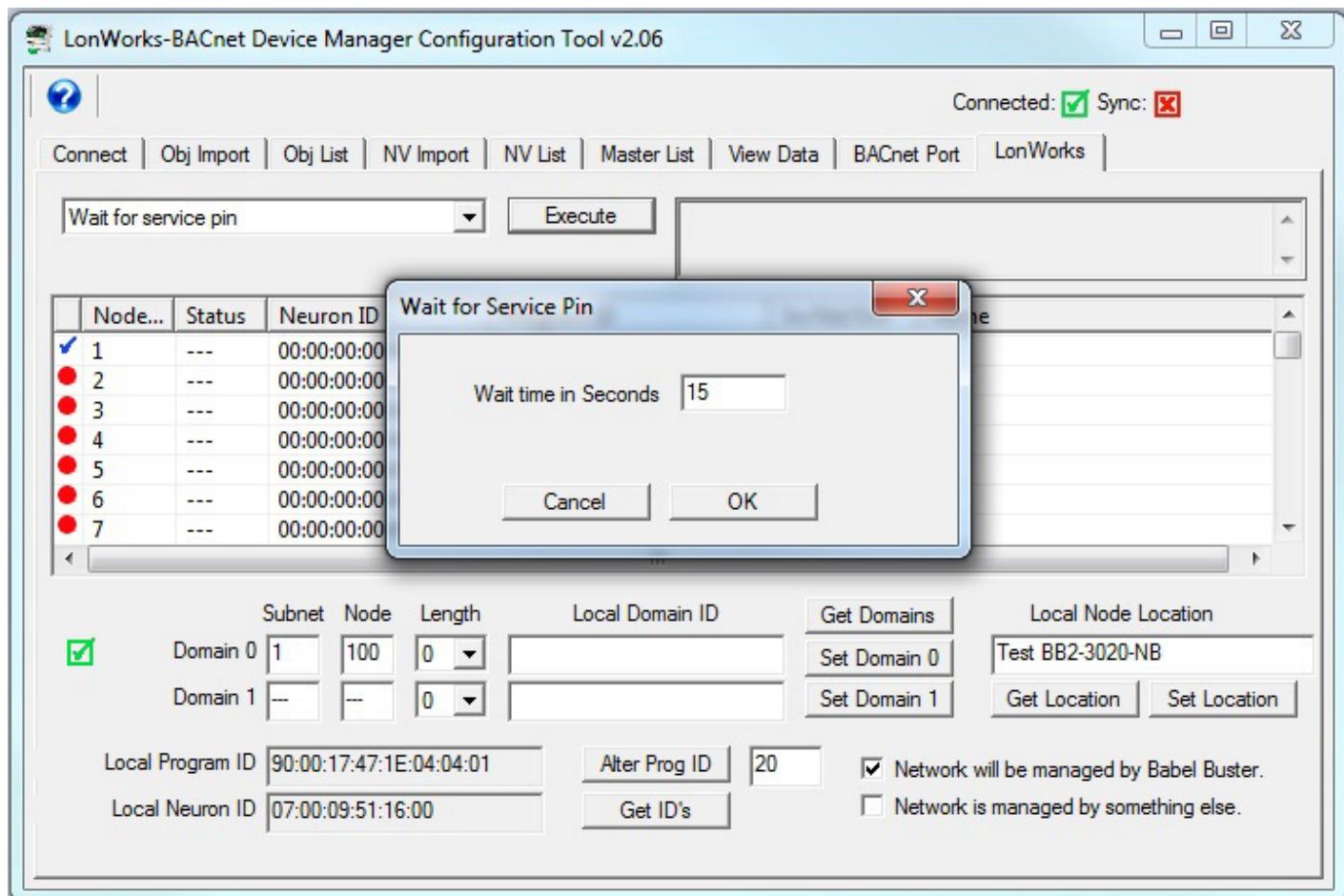
The selected node table entry will be indicated by a check mark in the icon column as illustrated here. (For discovery via network query, you can have multiple nodes showing as selected.)



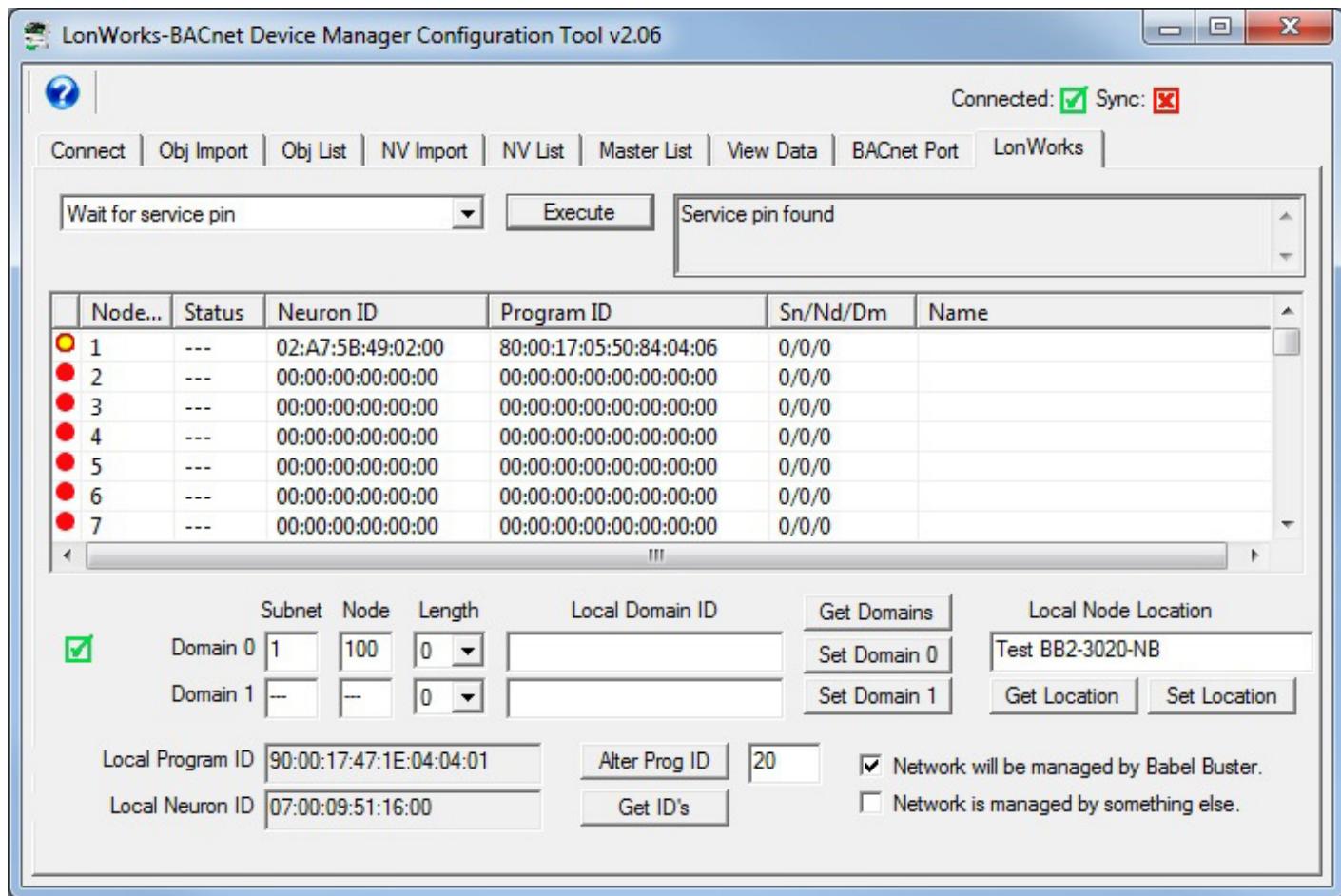
Provided you have a table entry selected, choose 'Wait for service pin' from the list and click Execute.



You will be given a dialog allowing you to specify a time period to wait for the service pin message to arrive on the LonWorks network (which is of course assumed to be connected to the device of interest). The wait time will default to 15 seconds, but if you have to hike a distance to get to the device, you will want to set this timeout much higher. Discovery via network query is preferred to avoid having to physically access the remote device, but sometimes the remote device has been previously configured in an incompatible manner and you have no choice.

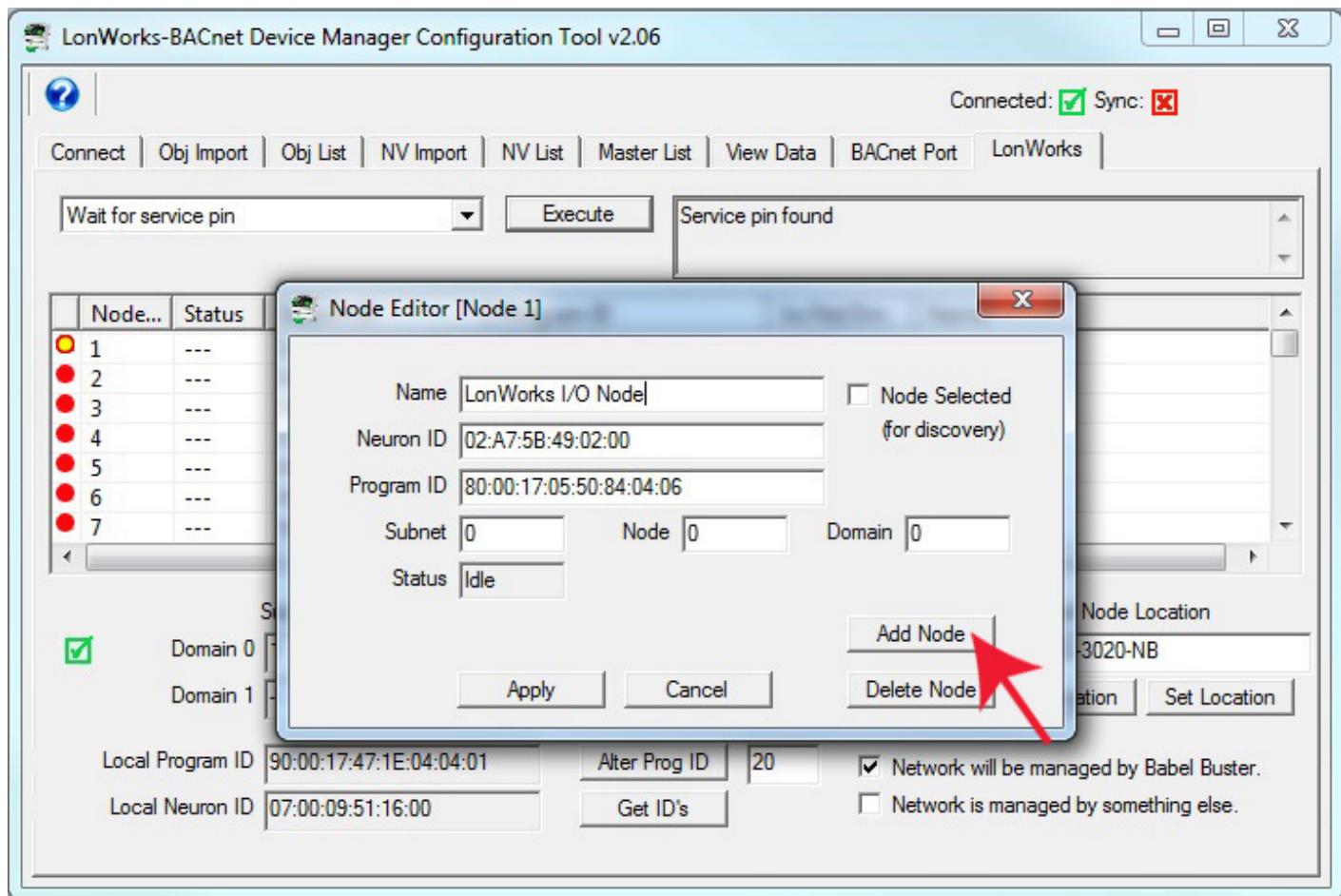


After clicking OK in the wait time dialog, proceed to press the button on the remote LonWorks device. When the service pin message is received, it will be indicated by an icon that is yellow with a red circle around it.

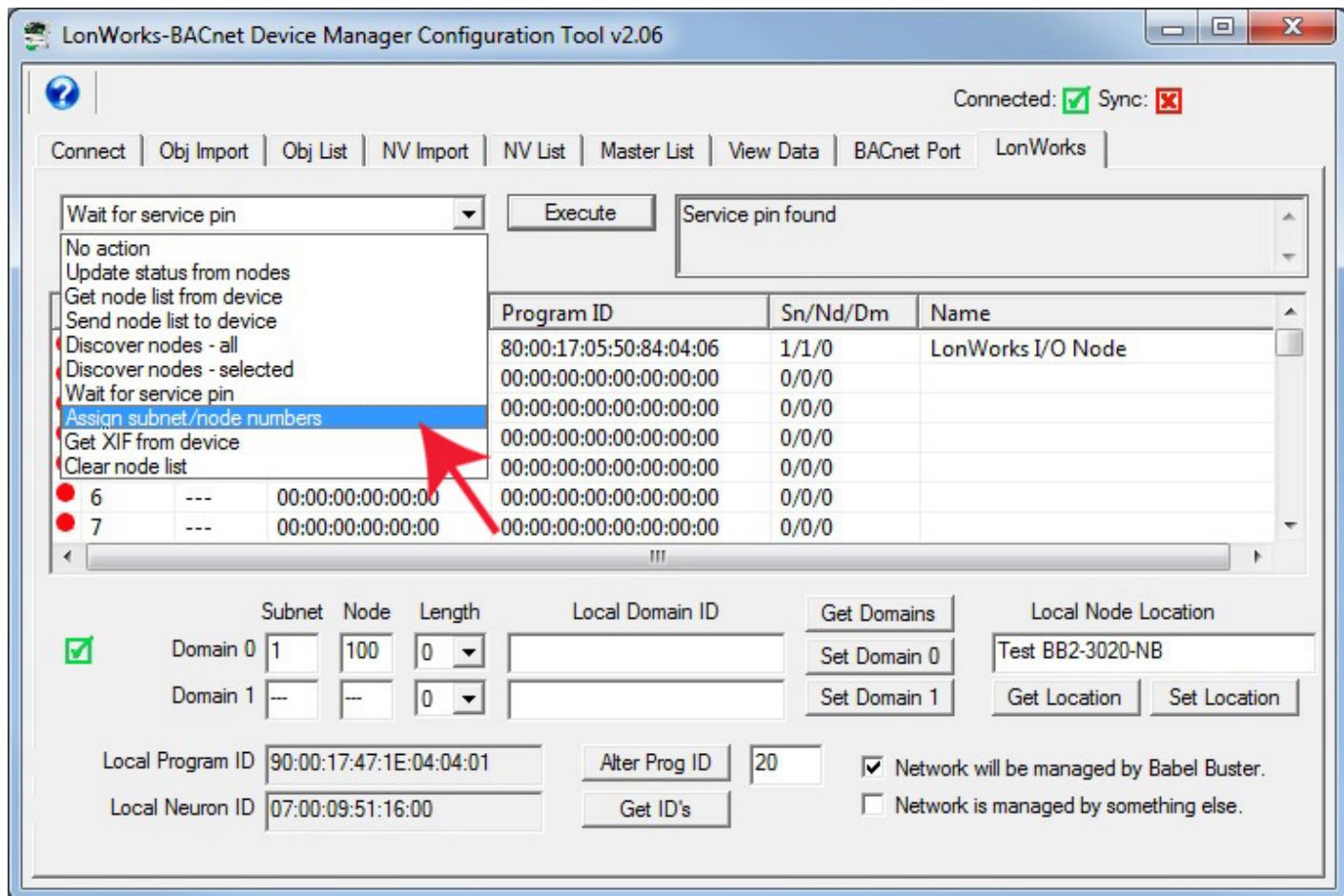


Next, you want to tell the tool 'yes, keep this one' by double clicking the node entry in the table to open the Node Editor dialog, and then clicking Add Node. You may also want to give the node a name at this point. Simply enter a name in the Name window before clicking Add Node. You can always come back to the Node Editor dialog later and add a name.

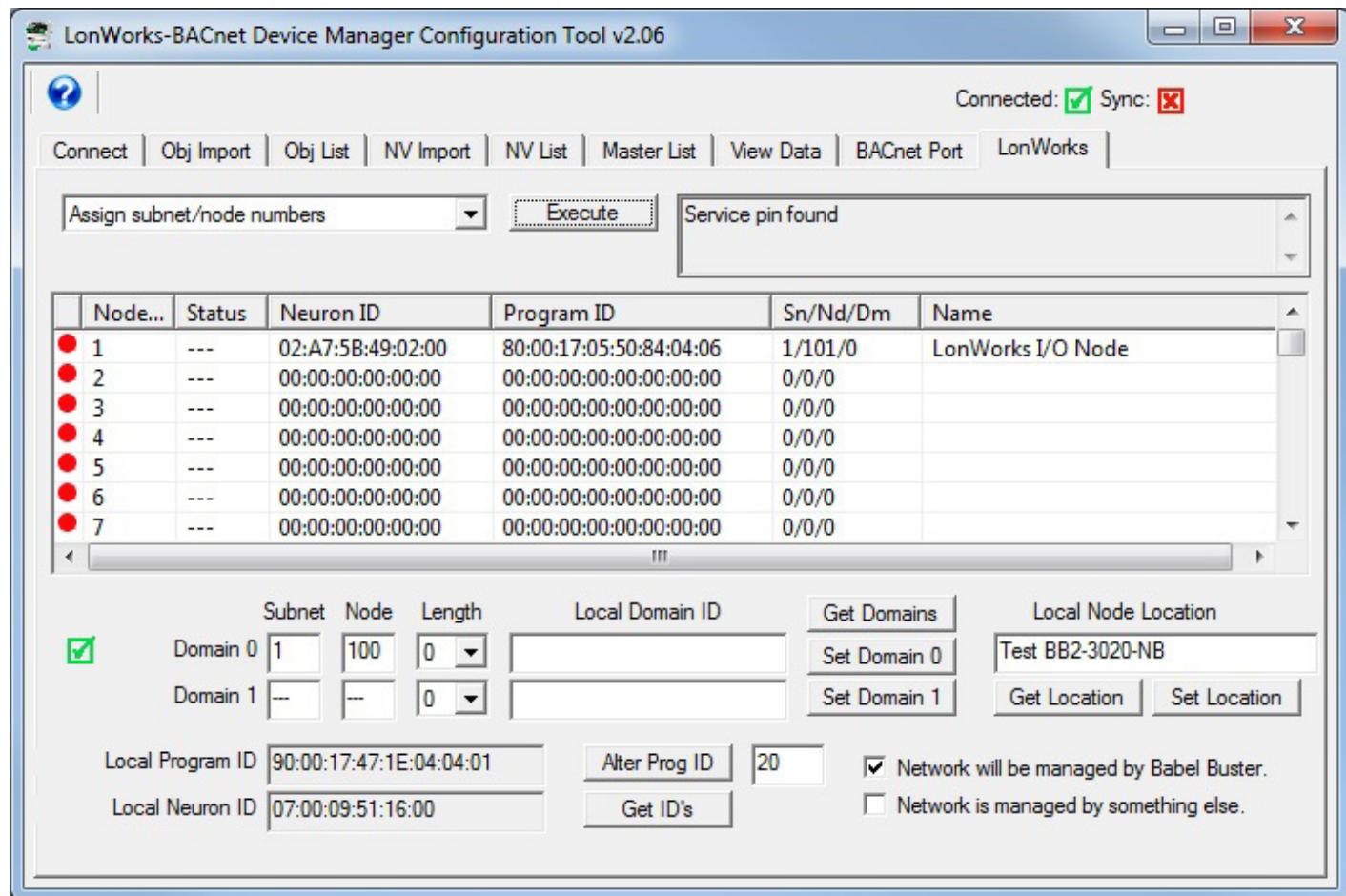
If you know exactly what subnet/node you want this device assigned to, you can also enter those numbers at this point. Be sure you know what you are doing here. If you use a subnet different than the gateway itself, you will not be able to communicate with it.



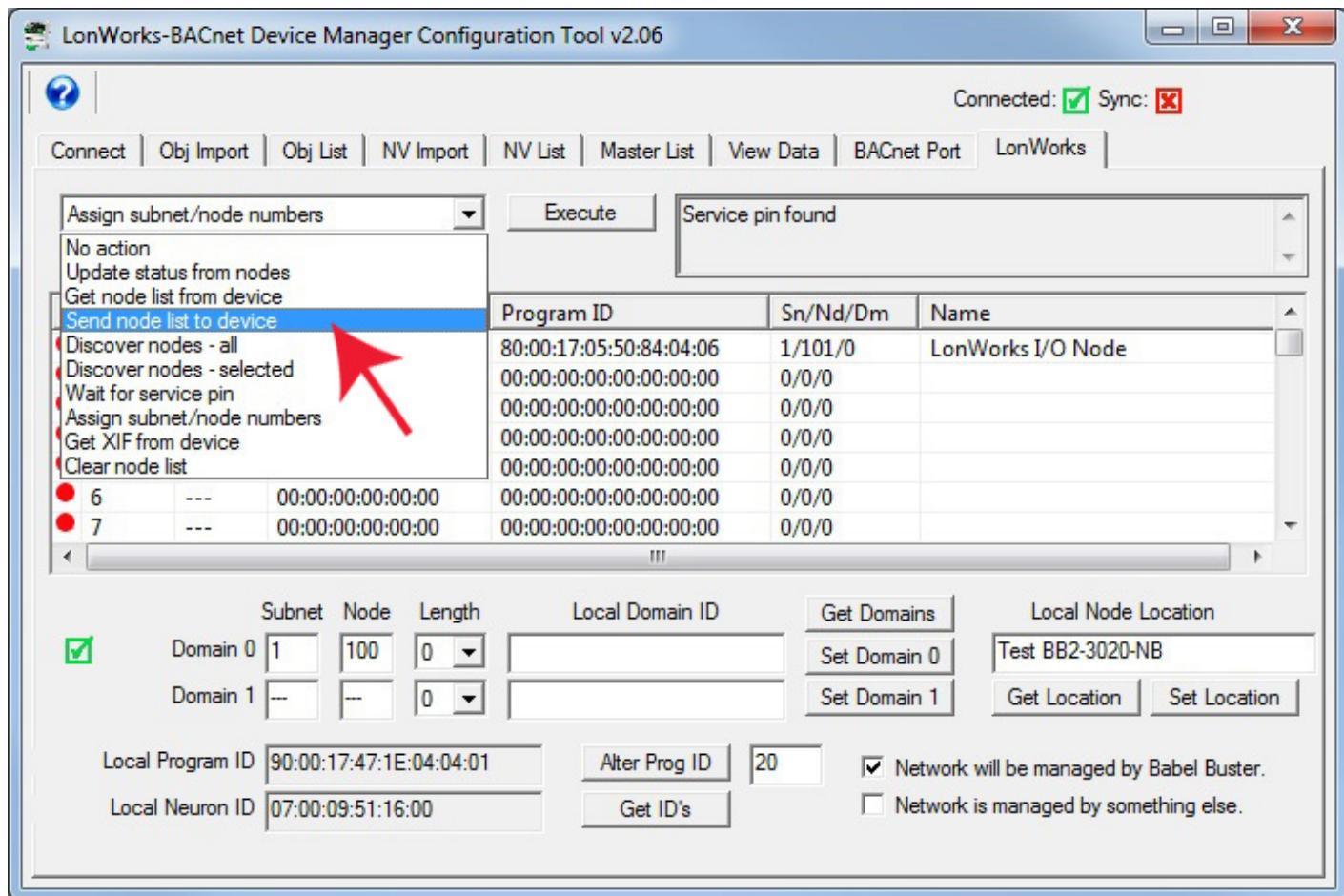
The best bet for picking subnet/node numbers is to let the configuration tool do it for you. Select 'Assign subnet/node numbers' from the list and click Execute.



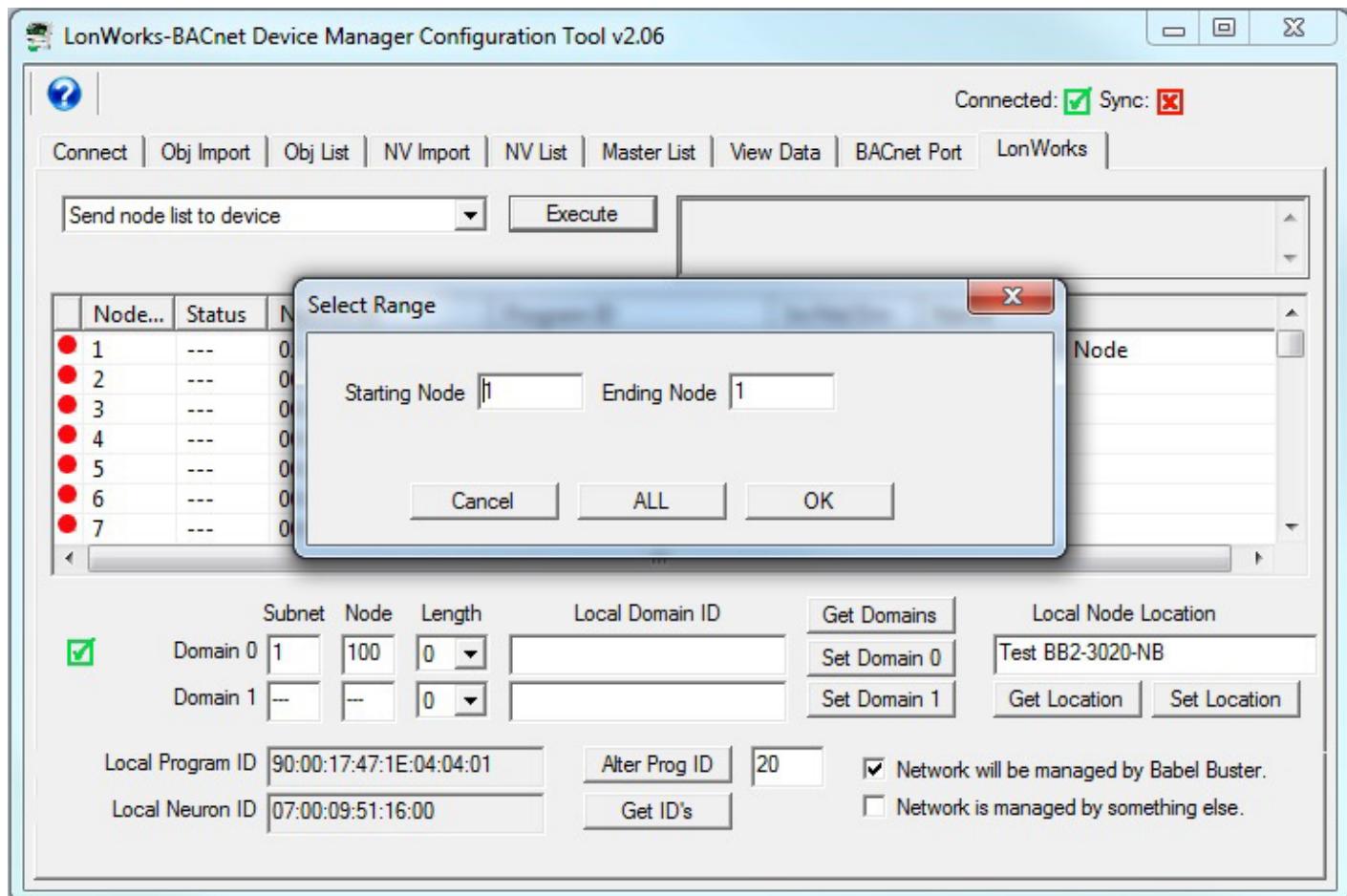
The gateway will always assign its own subnet, and will assign node numbers incrementing upward from its own node number plus one. Once you have 'added' the node and assigned a subnet/node, the icon in the first column will change to solid red indicating this is now a valid node table entry that has not yet been sent to the gateway. (Receiving a service pin message does not populate the node table in the gateway device, it only populates the table in the configuration tool.)



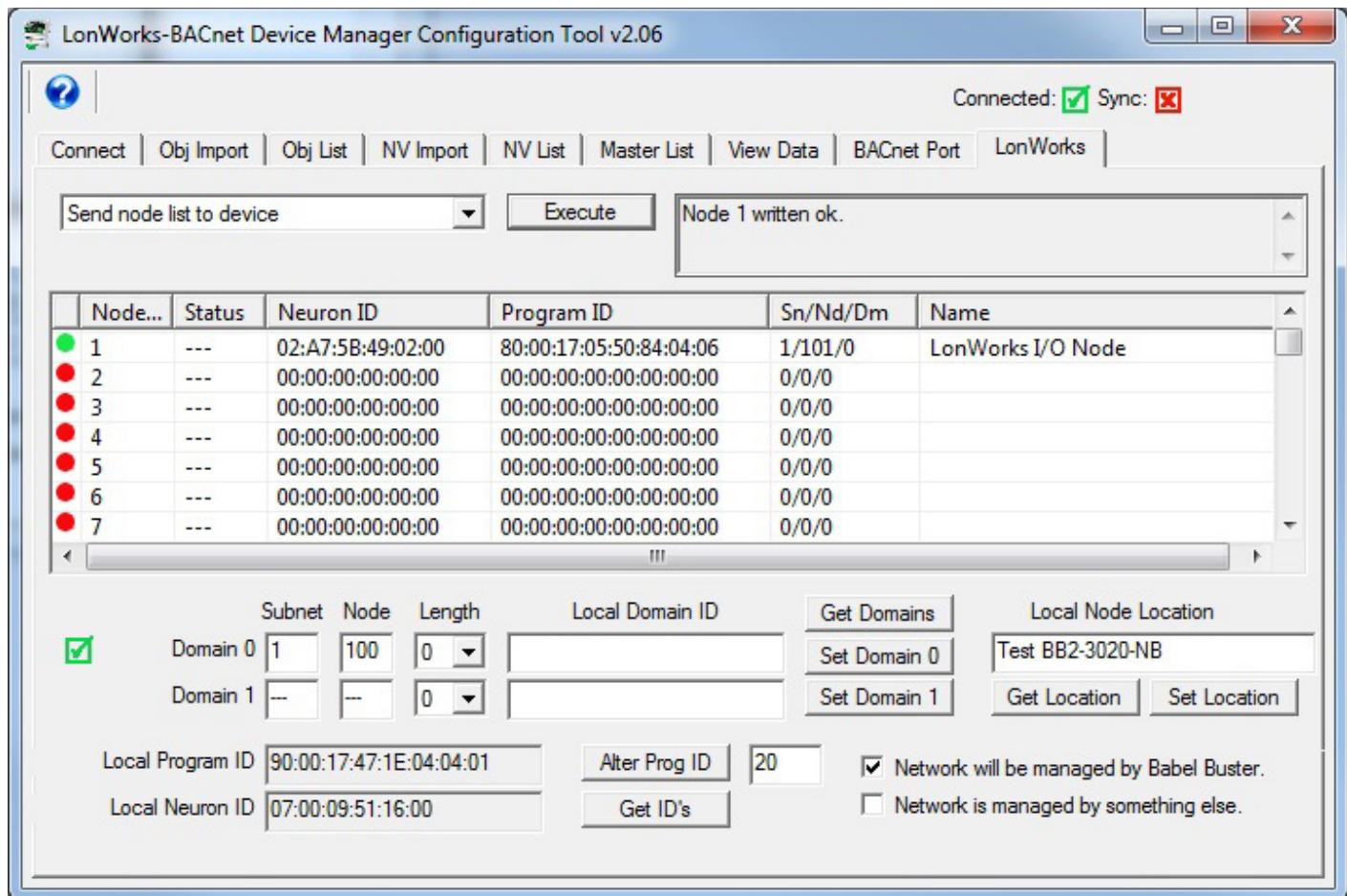
To add this node to the node table in the gateway device, select 'Send node list to device' and click Execute.



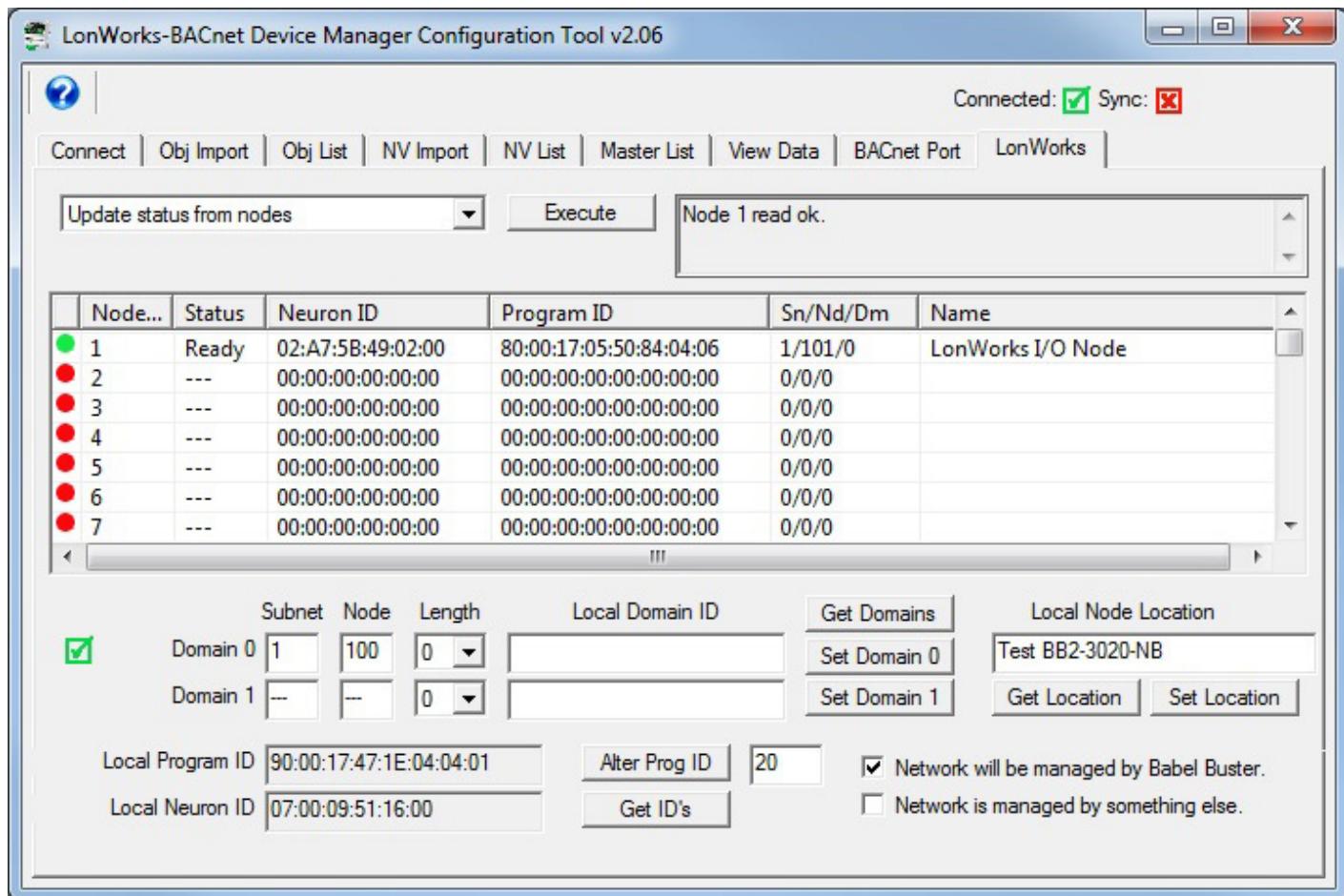
Any time you send the node list to the device (or get it from the device), you will be given the option of selecting a range. When sending node table entries to the gateway, it will default to a range of those nodes that are configured in the tool but not configured yet in the device.



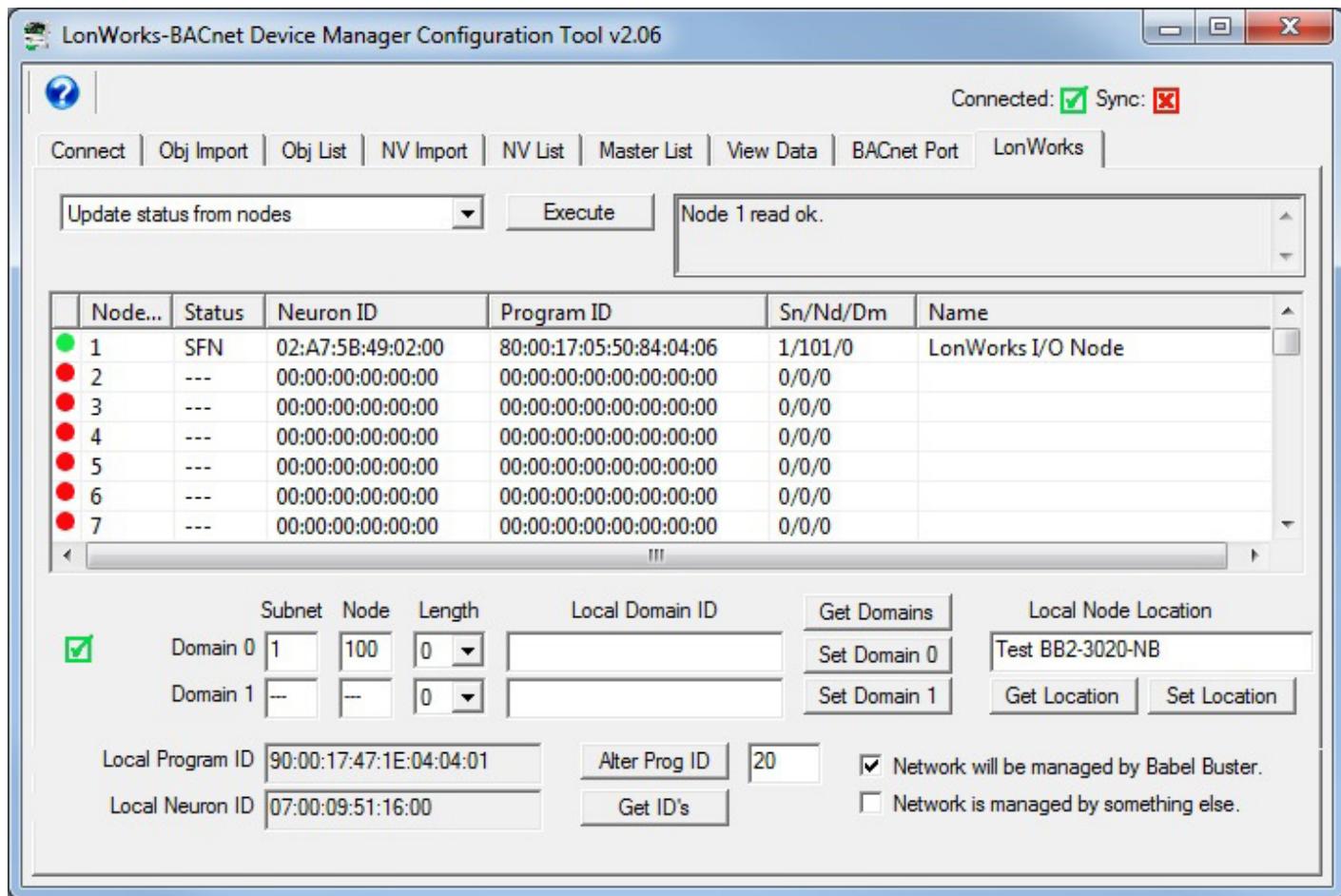
Once the node table entry is sent to the device, its icon will become green.



Next, you will want to select 'Update status from nodes' and click Execute (and provide range or just click OK in range dialog). If the node was just recently sent to the gateway, its status might not be "Ready" immediately. Try again in a few moments if not ready. If status remains not ready for an extended period, the status will indicate where it is stalled.



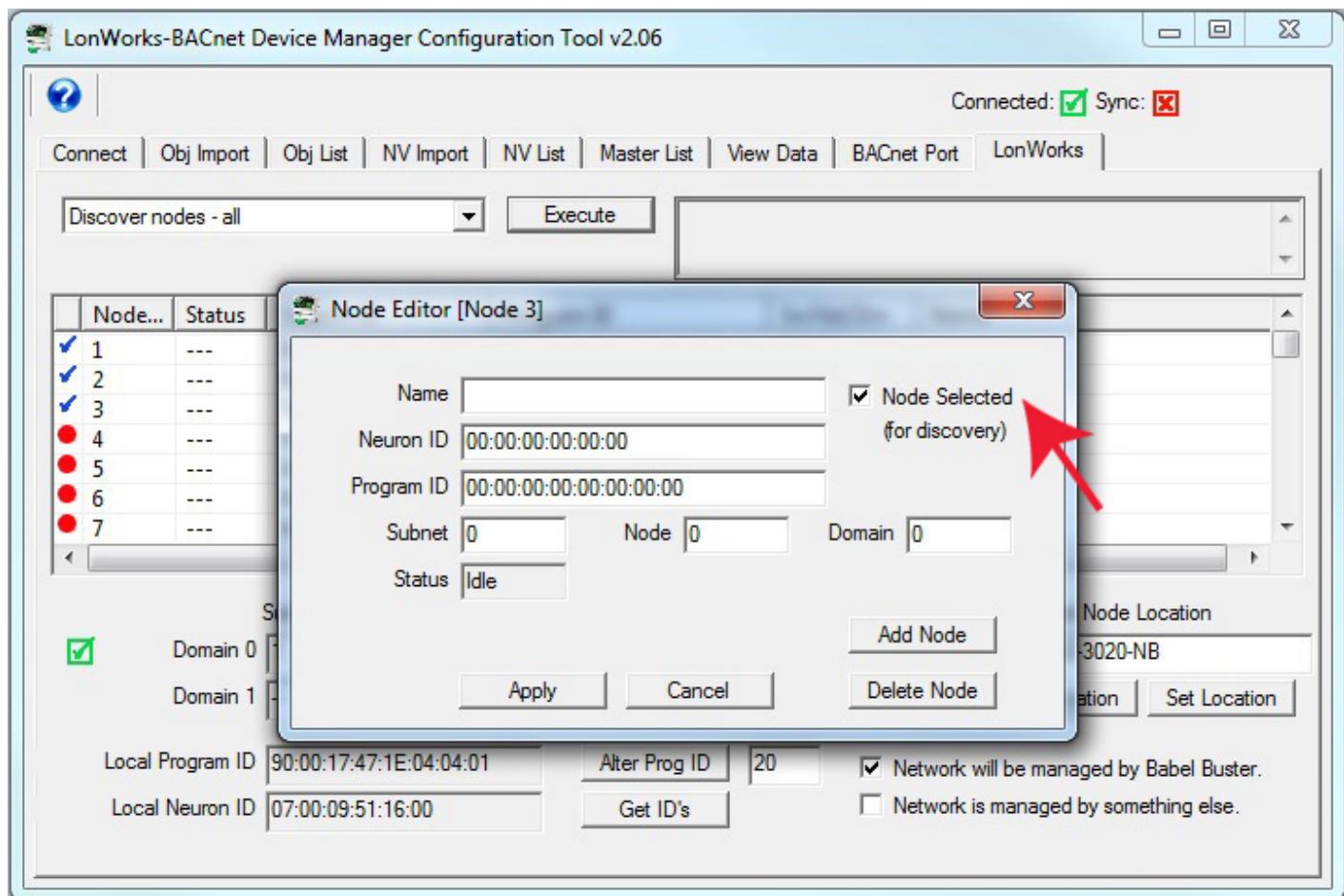
The most common non-ready status will be SFN. This means it was unable to force the remote LonWorks device onto the same domain as the gateway. It will retry every 15 seconds until successful, or forever if not.



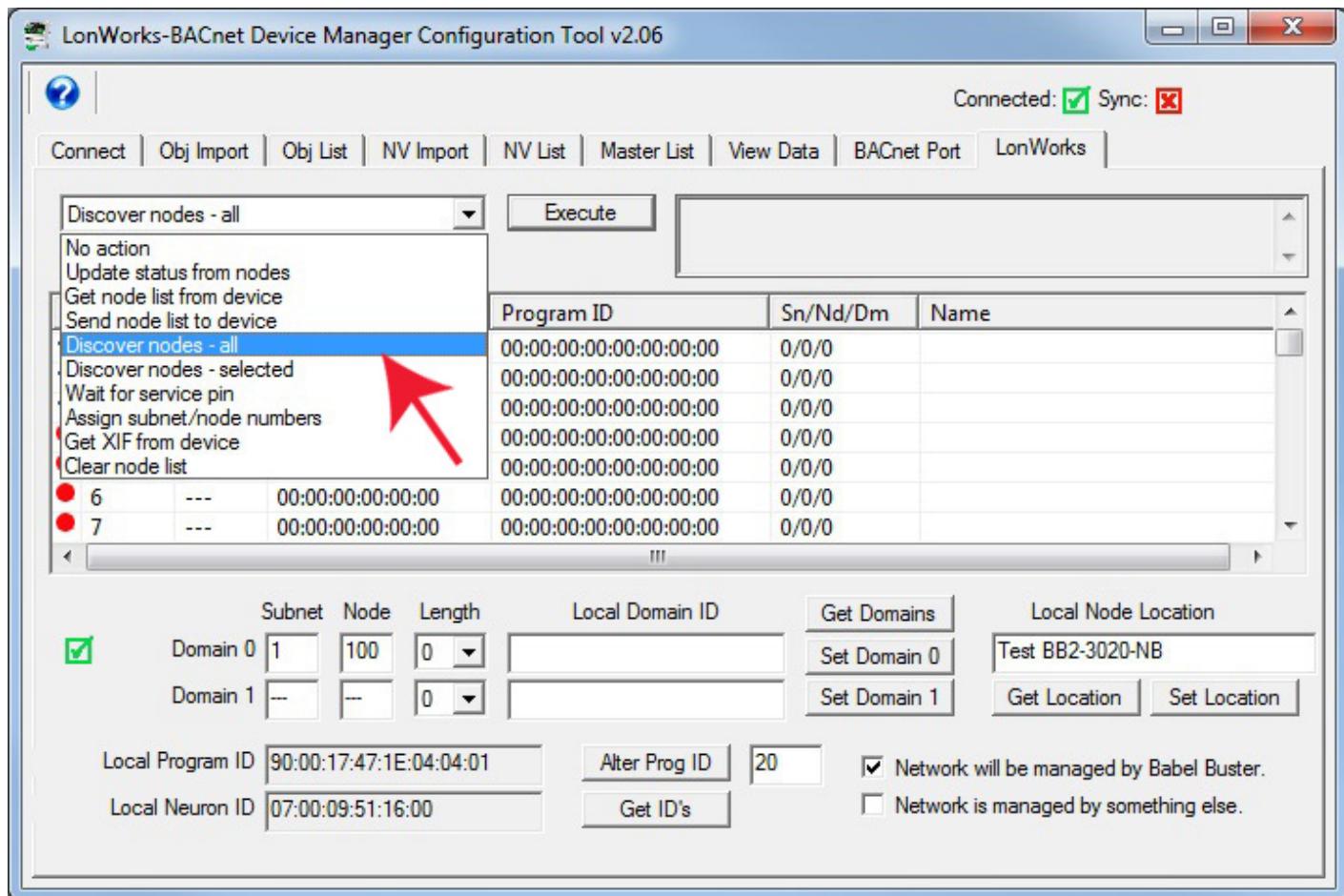
12.5 Discovery of All Nodes via Network Query

Discovery via network query is easy in that you do not need to physically access the remote LonWorks device. However, it does require that the remote device's domain table is configured such that the gateway is able to query it. If the domain length is zero, then the ID is effectively a wild card. LonWorks protocol recommends that all LonWorks products be manufactured with a default domain length of zero. But if the device had been previously commissioned on a managed network, the network management tool most likely changed the domain to something unknown to (and incompatible with) the Babel Buster gateway. You then have two options: (a) Use the service pin method outlined above, or (b) Find out what domain the previous network management tool used and change the gateway's domain to match it.

To begin the discovery process, double click a node table entry to open the Node Editor dialog, and click Node Selected. Do this for as many nodes as you want to discover so that multiple unused entries in the table show a check mark in the icon column.

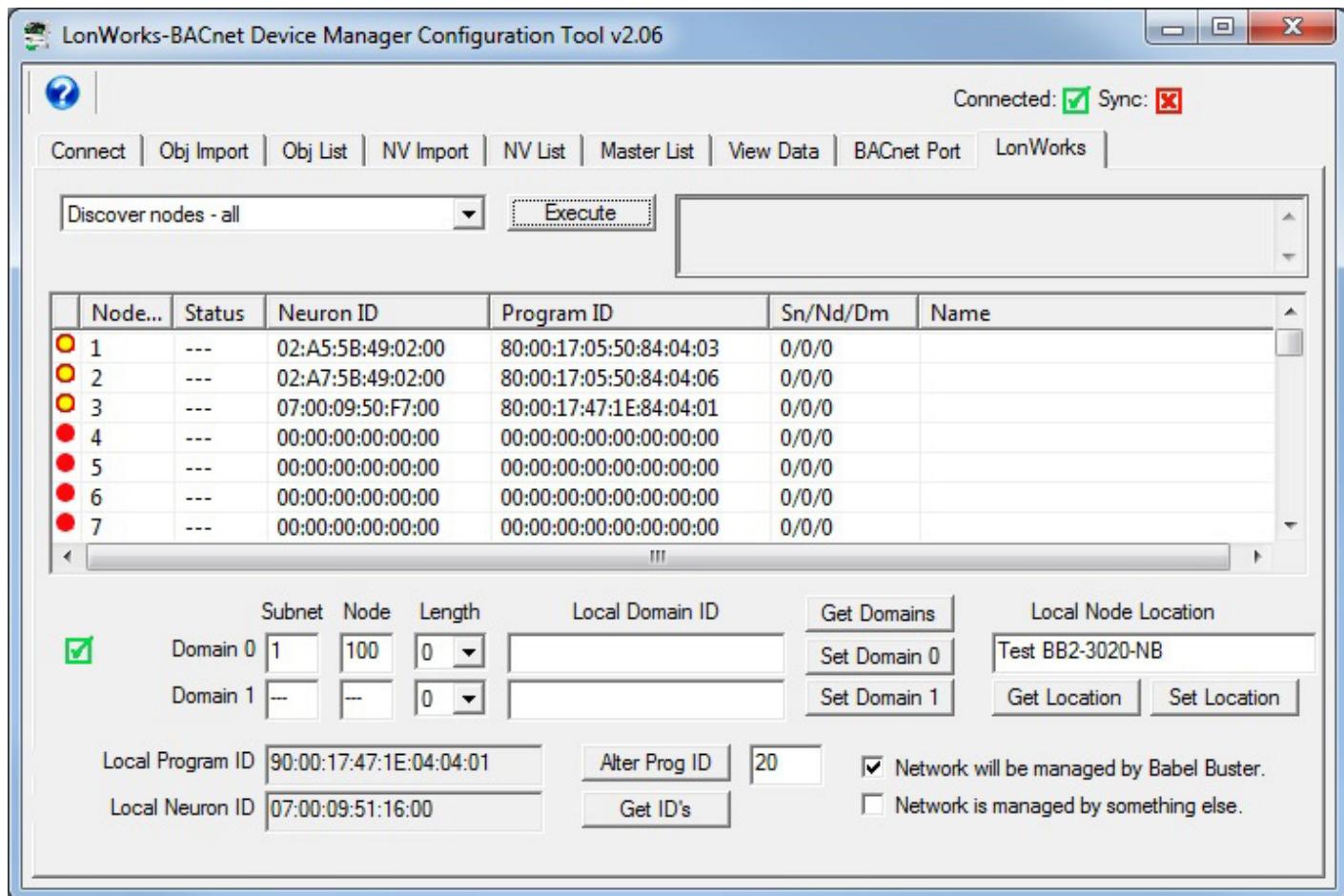


Select 'Discover nodes - all' from the list and click Execute.



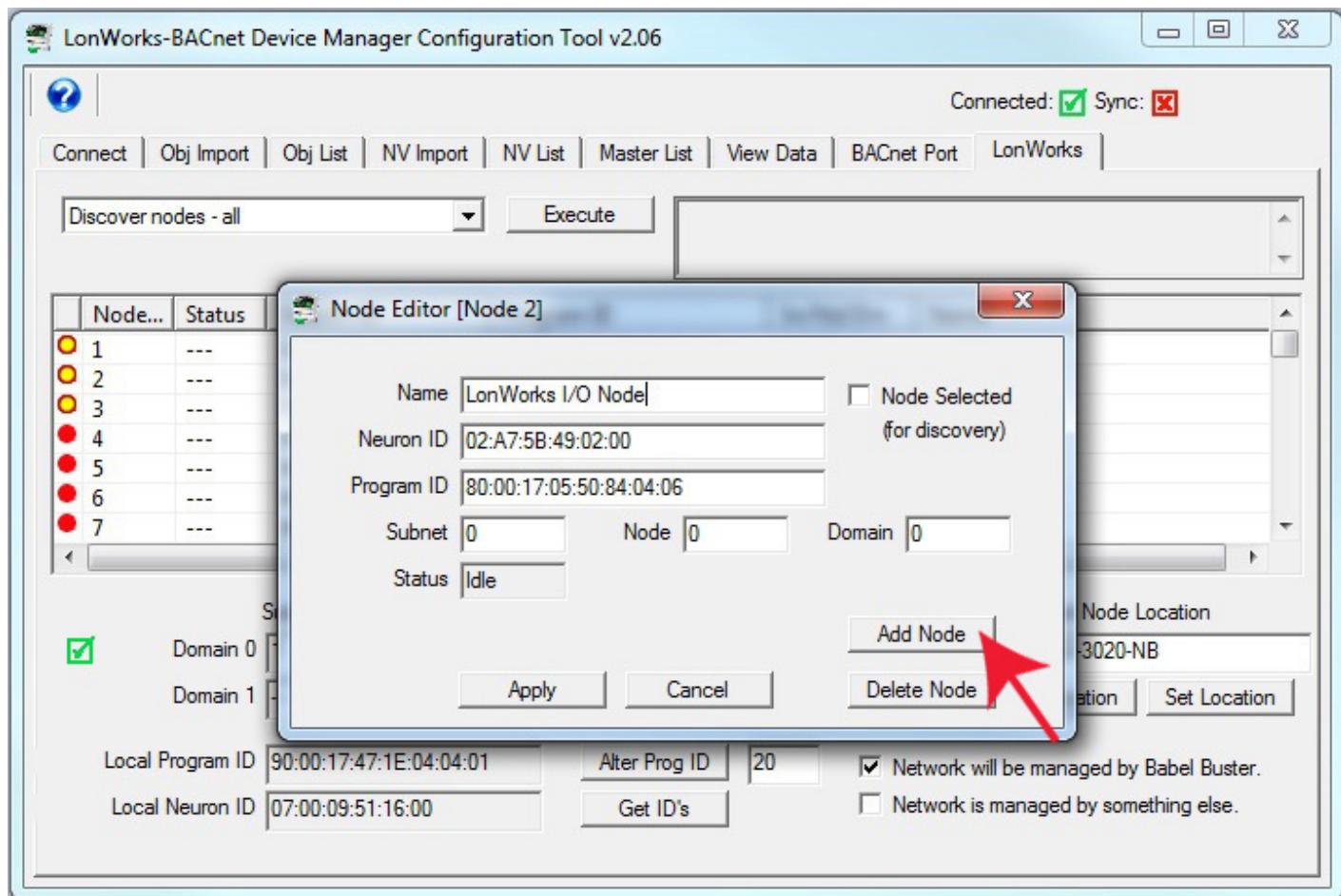
The gateway now sends out a query to the network. This process has a very specific technical definition, but in general terms, what is happening is this: The gateway announces to the world, "Hey, if you're out there, talk to me, and tell me who you are." If anything on the LonWorks network was able to hear that announcement, it attempts to reply with "Here I am, and this is my ID".

For each response received by the gateway, it will begin to fill in the selected node table entries in the tool. The icon will change to yellow with a red circle indicating that this is a discovered node. The node table in the gateway itself is not yet populated. Only the table in the configuration tool is being populated by this process.



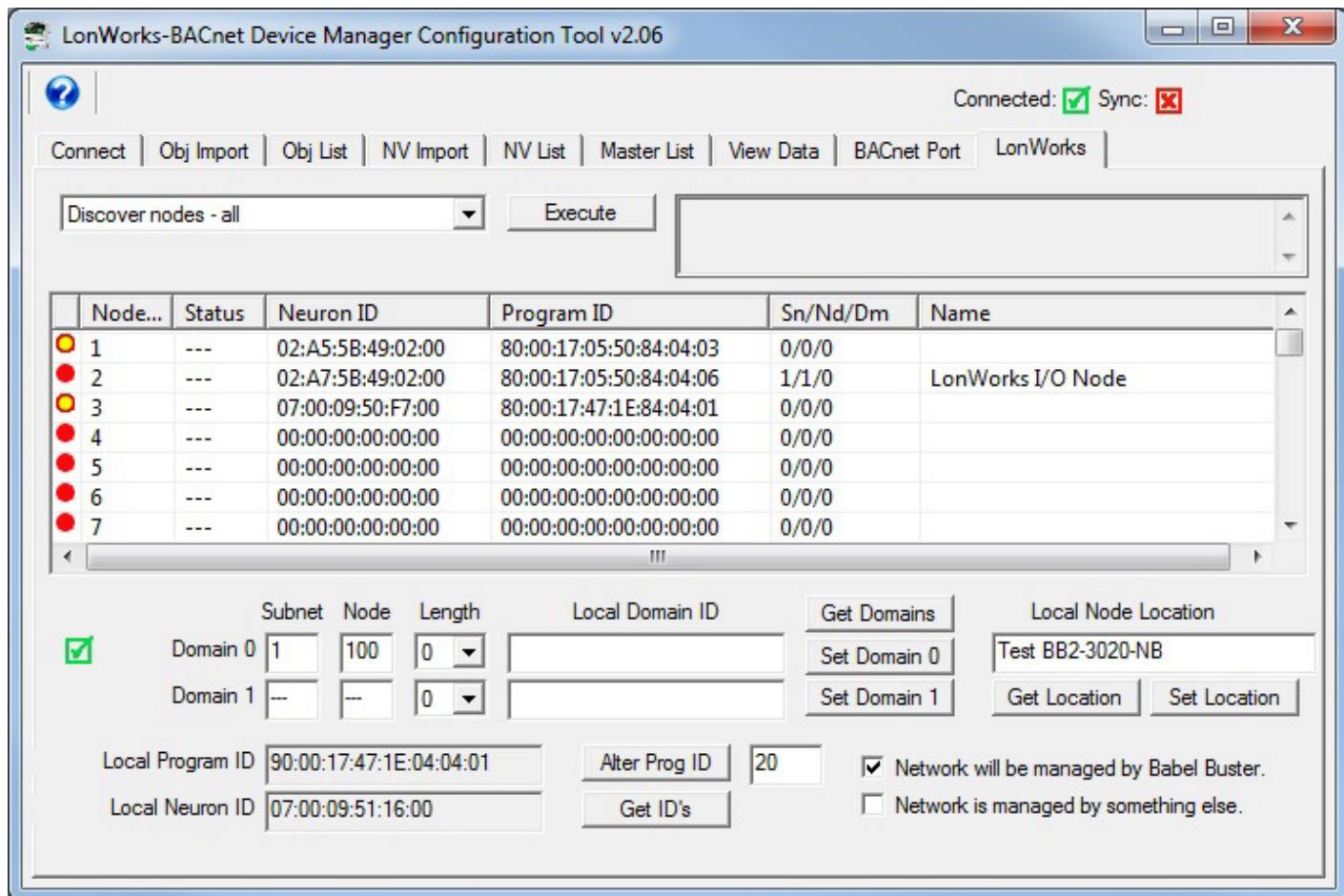
To "keep" the discovered node, double click on it, and click Add Node in the Node Editor. You may also want to give the node a name at this point. Simply enter a name in the Name window before clicking Add Node. You can always come back to the Node Editor dialog later and add a name.

If you know exactly what subnet/node you want this device assigned to, you can also enter those numbers at this point. Be sure you know what you are doing here. If you use a subnet different than the gateway itself, you will not be able to communicate with it.

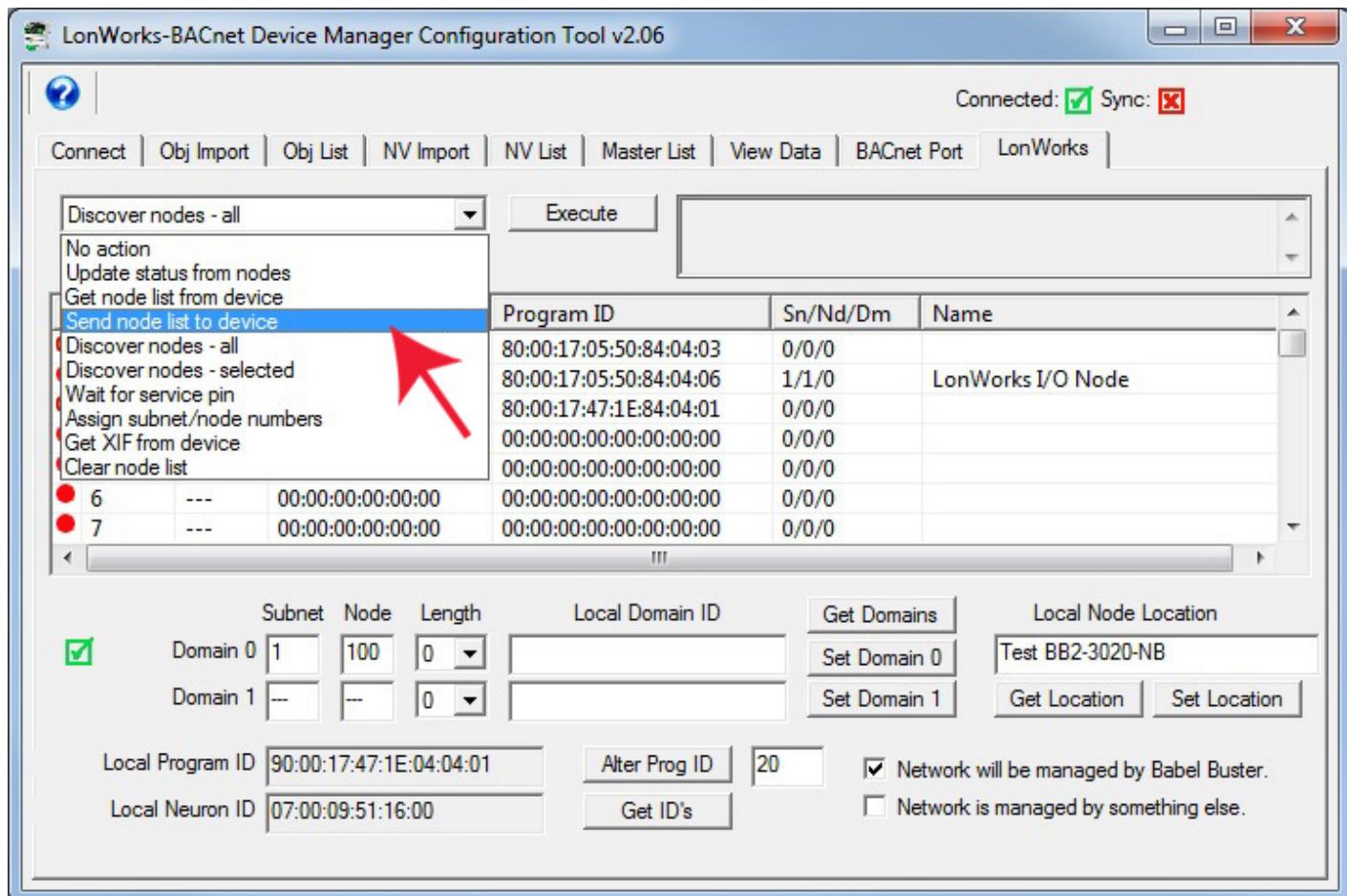


In this example, we have manually entered subnet 1, node 1, prior to clicking Add Node. You should use the same subnet as the gateway itself, and any node number in the range of 1 to 127 that is not used by the gateway itself, or any other node already in the table. Better yet, simply select 'Assign subnet/node numbers' from the list and click Execute - but wait! Don't do this until you have either added all nodes, and/or deleted those you do not care about.

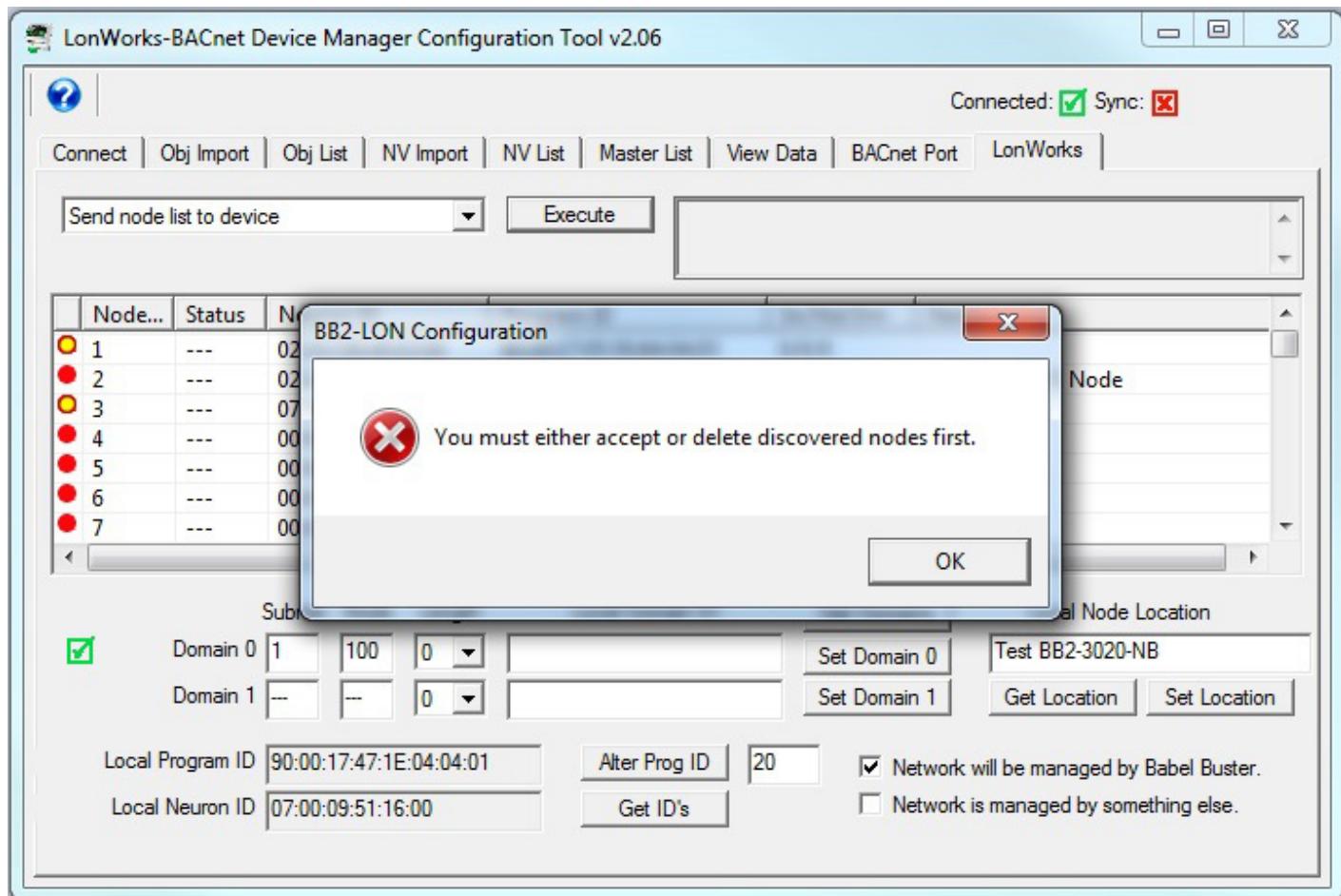
After entering a name and subnet/node, and then clicking Add Node, our table now looks like this.



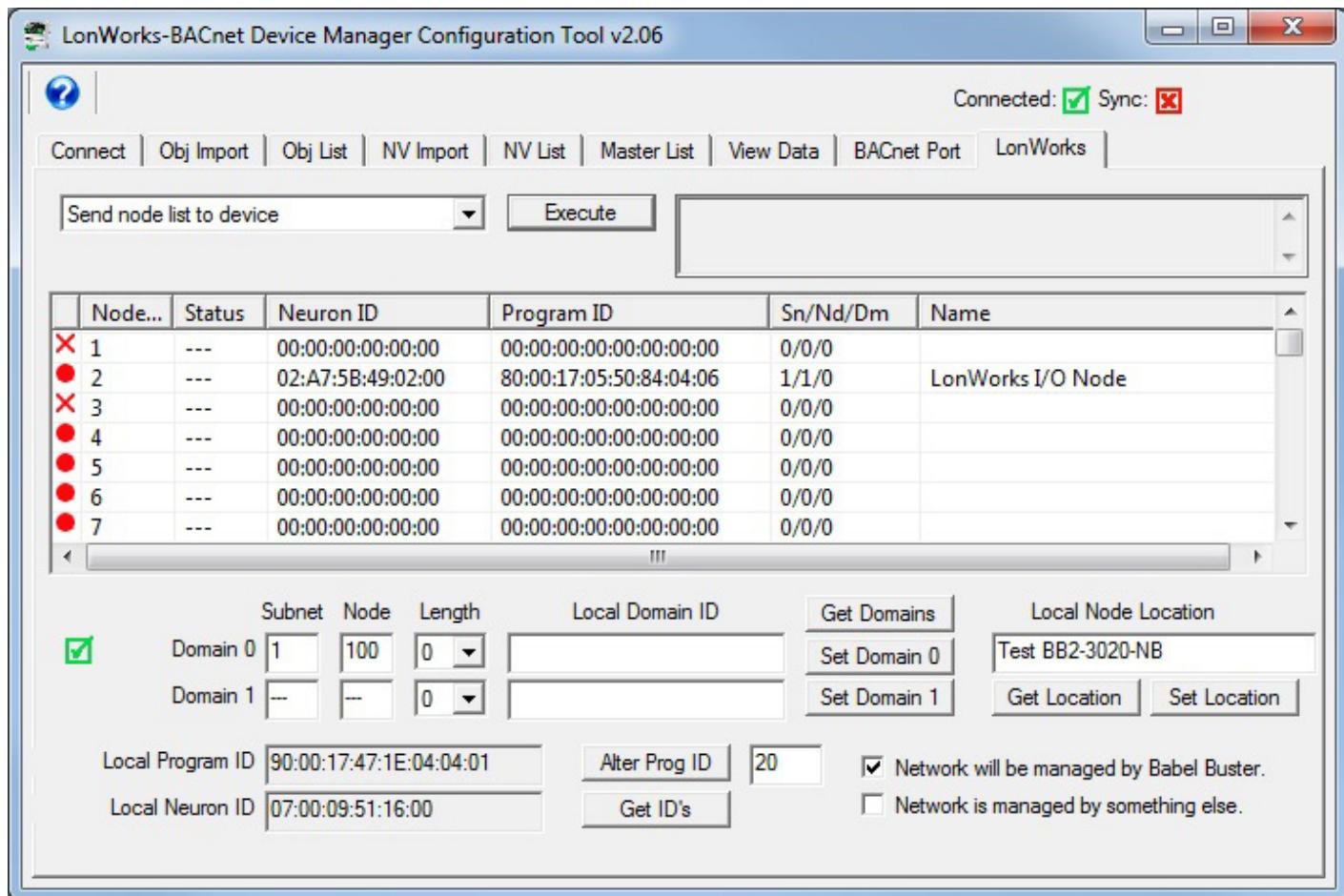
The node table is only populated in the configuration tool at this point. To populate the node table in the Babel Buster gateway, select 'Send node list to device' and click Execute.



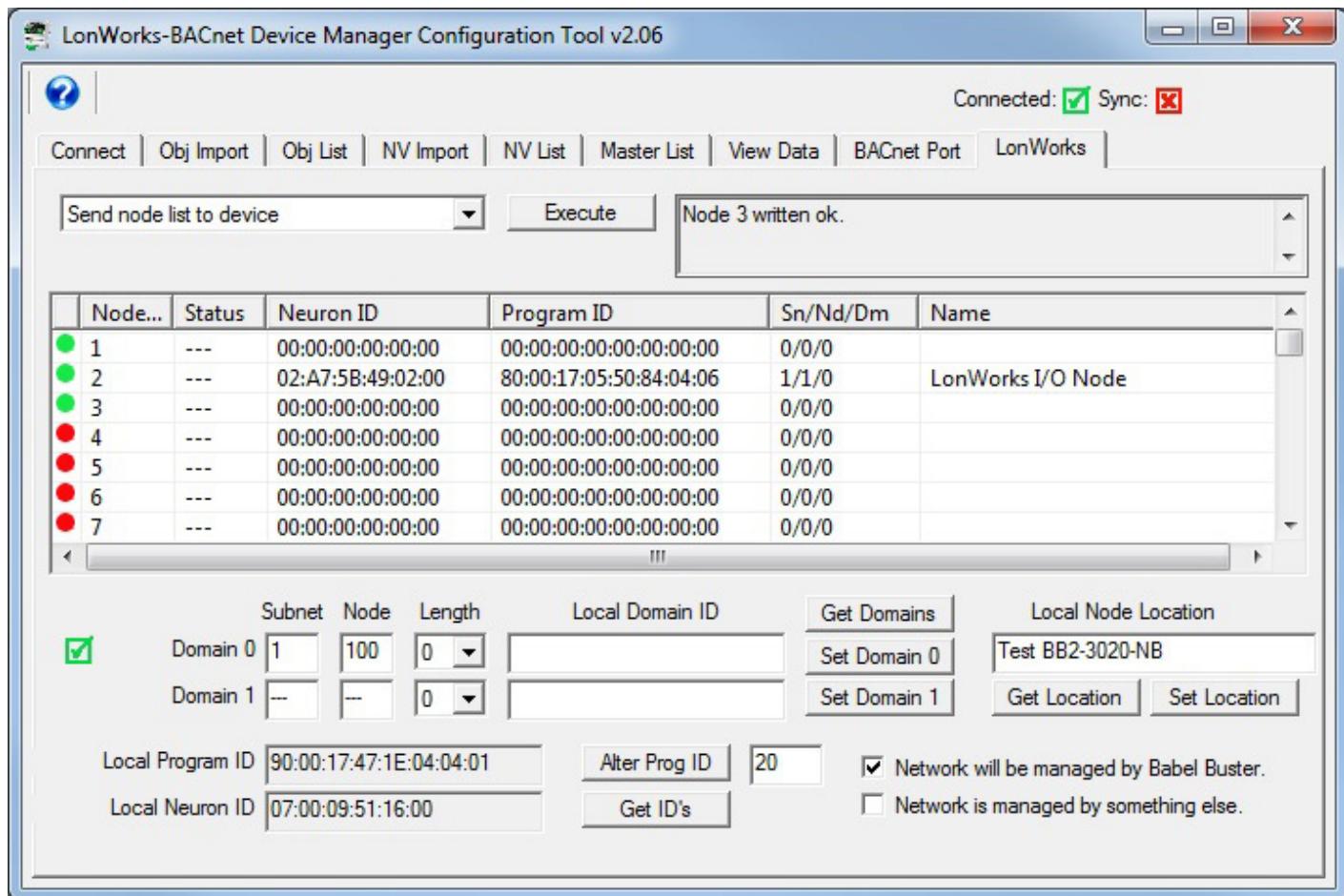
But there's a problem we deliberately left in place here to demonstrate. You need to either accept or reject all of the discovered nodes before you can send the new node table to the gateway device.



To get rid of a node table entry, double click that entry, and instead of clicking Add Node, click Delete Node. The deleted node will be indicated by a red X in the icon column.



You can now send the cleaned up list to the gateway device. The green icon indicates that this node table entry now exists in the Babel Buster gateway device, not just in the configuration tool.



12.6 Discovery of Selected Node via Network Query

If you know a little bit about the remote LonWorks device you wish to discover and want to be sure you only discover that type of device, there is a method of accomplishing this. This process is especially useful if there is a large number of LonWorks devices on the network and you don't care about most of them.

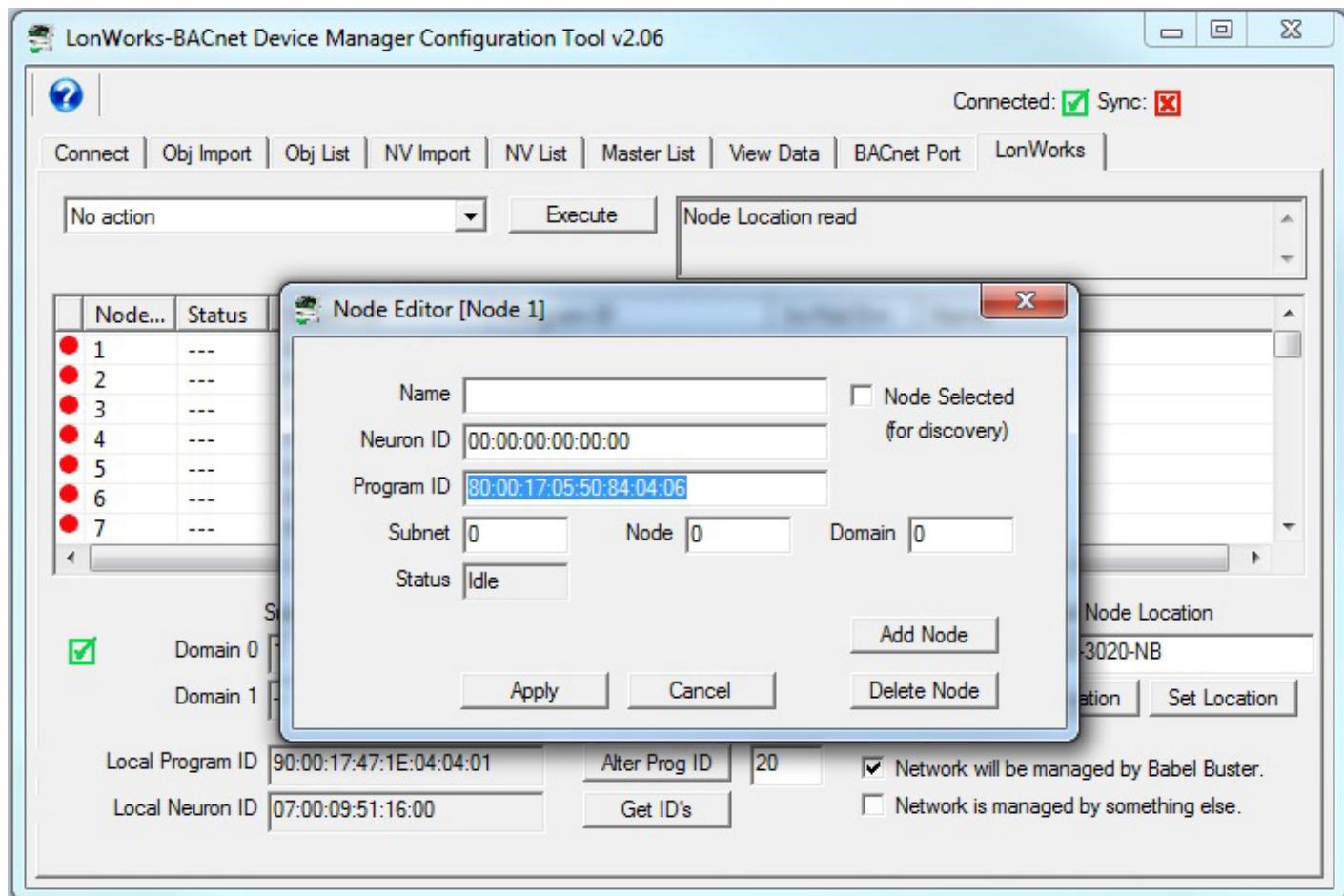
You need to obtain the Program ID of the device(s) of interest. If you imported an XIF file earlier, the program ID found in that file appears on the NV Import page. Simply copy that text from the NV Import page.

The screenshot shows the LonWorks-BACnet Device Manager Configuration Tool v2.06. The interface includes a menu bar with File, Edit, View, Tools, Help, and a toolbar with icons for Connect, Import, Export, and Help. The main window has tabs for Connect, Obj Import, Obj List, NV Import, NV List, Master List, View Data, BACnet Port, and LonWorks. The NV Import tab is selected, showing a message "11 NV's parsed from AM2D6.XIF." and fields for "Add to NV List" (checkbox checked), "Sync" (checkbox checked), "Assign to Node # 1", "Program ID 80:00:17:05:50:84:04:06", and "Default Poll Time 15". Below these are two tables: one for NV entries and one for Master List entries.

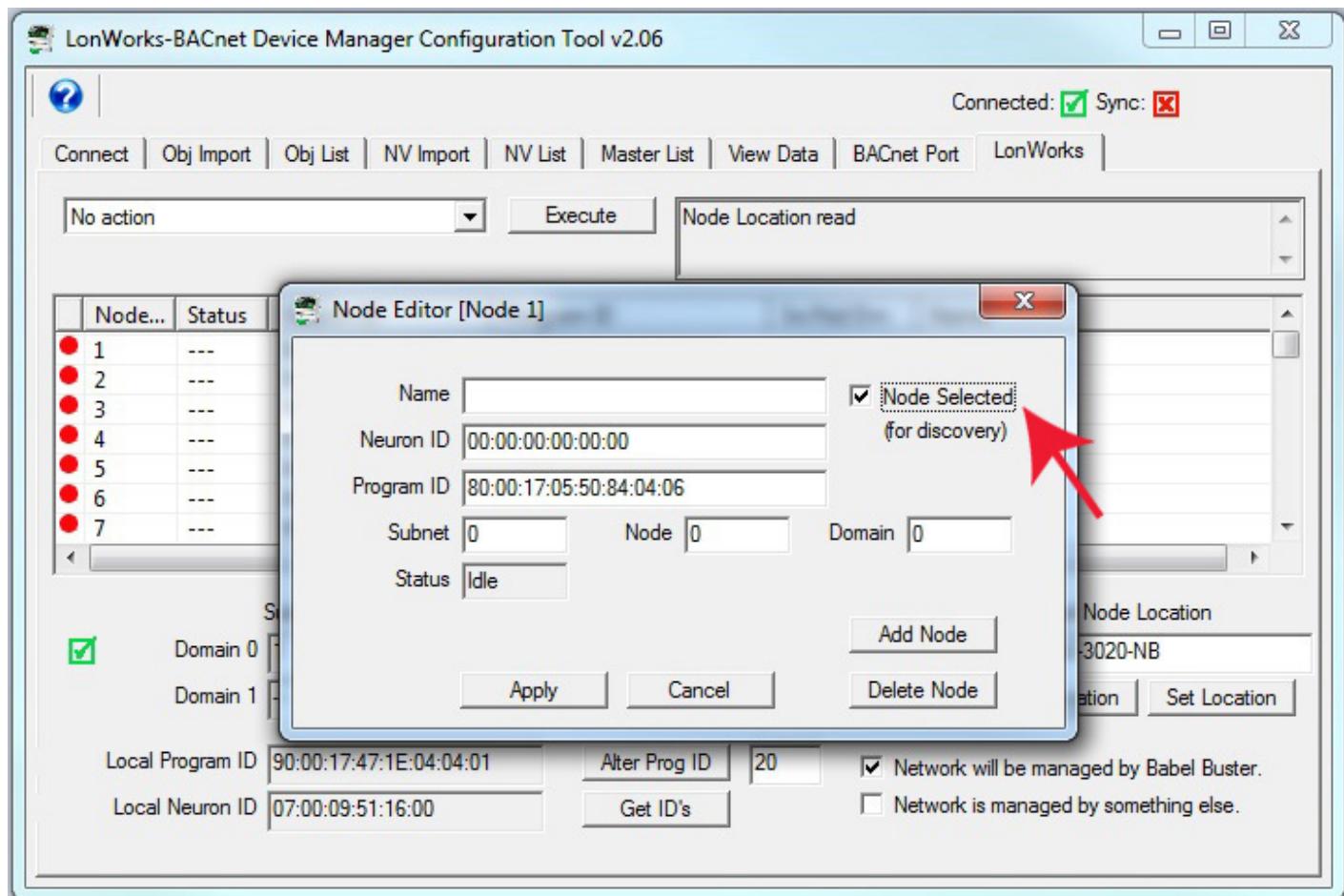
Dir	Index	SNVT Type	Name
NVI	0	SNVT_obj_request	nviRequest
NVI	1	SNVT_time_stamp	nviTimeSet
NVO	2	SNVT_obj_status	nvoStatus
NVO	3	SNVT_address	nvoFileDirectory
NVO	4	SNVT_alarm2	nvoAlarm2
NVO	5	SNVT_volt_f	nvoAnalogIn_1
NVO	6	SNVT_volt_f	nvoAnalogIn_2
NVO	7	SNVT_volt_f	nvoAnalogIn_3
NVO	8	SNVT_volt_f	nvoAnalogIn_4
NVI	9	SNVT_switch	nviDiscreteOut_1
NVI	10	SNVT_switch	nviDiscreteOut_2

Index	SNVT Type	Name
1	SNVT_time_stamp	nviTimeSet
2	SNVT_obj_status	nvoStatus
3	SNVT_address	nvoFileDirectory
4	SNVT_alarm2	nvoAlarm2
5	SNVT_volt_f	nvoAnalogIn_1
6	SNVT_volt_f	nvoAnalogIn_2
7	SNVT_volt_f	nvoAnalogIn_3
8	SNVT_volt_f	nvoAnalogIn_4
9	SNVT_switch	nviDiscreteOut_1
10	SNVT_switch	nviDiscreteOut_2

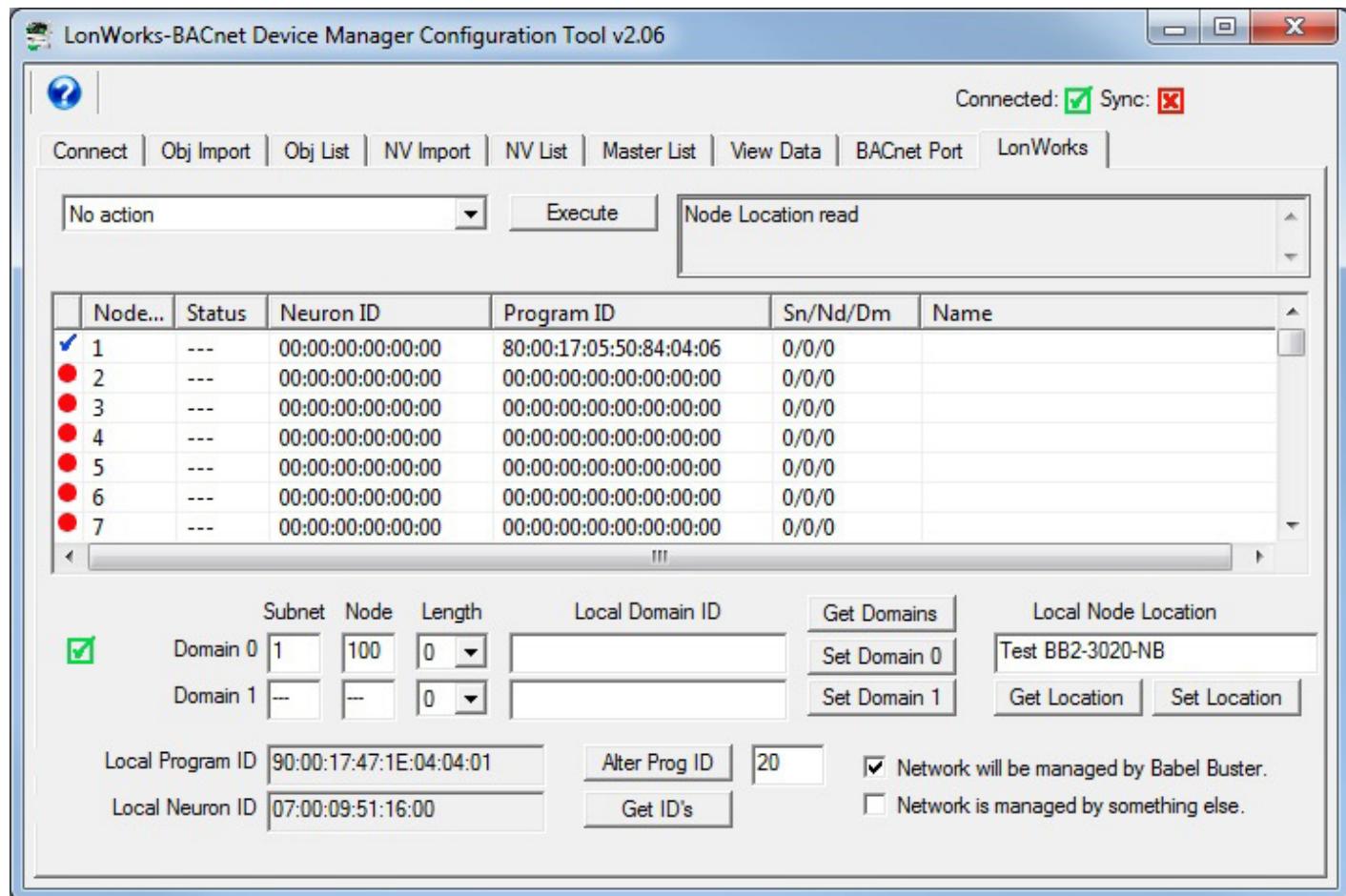
Double click an unused entry in the node table, and paste the Program ID into the Program ID window.



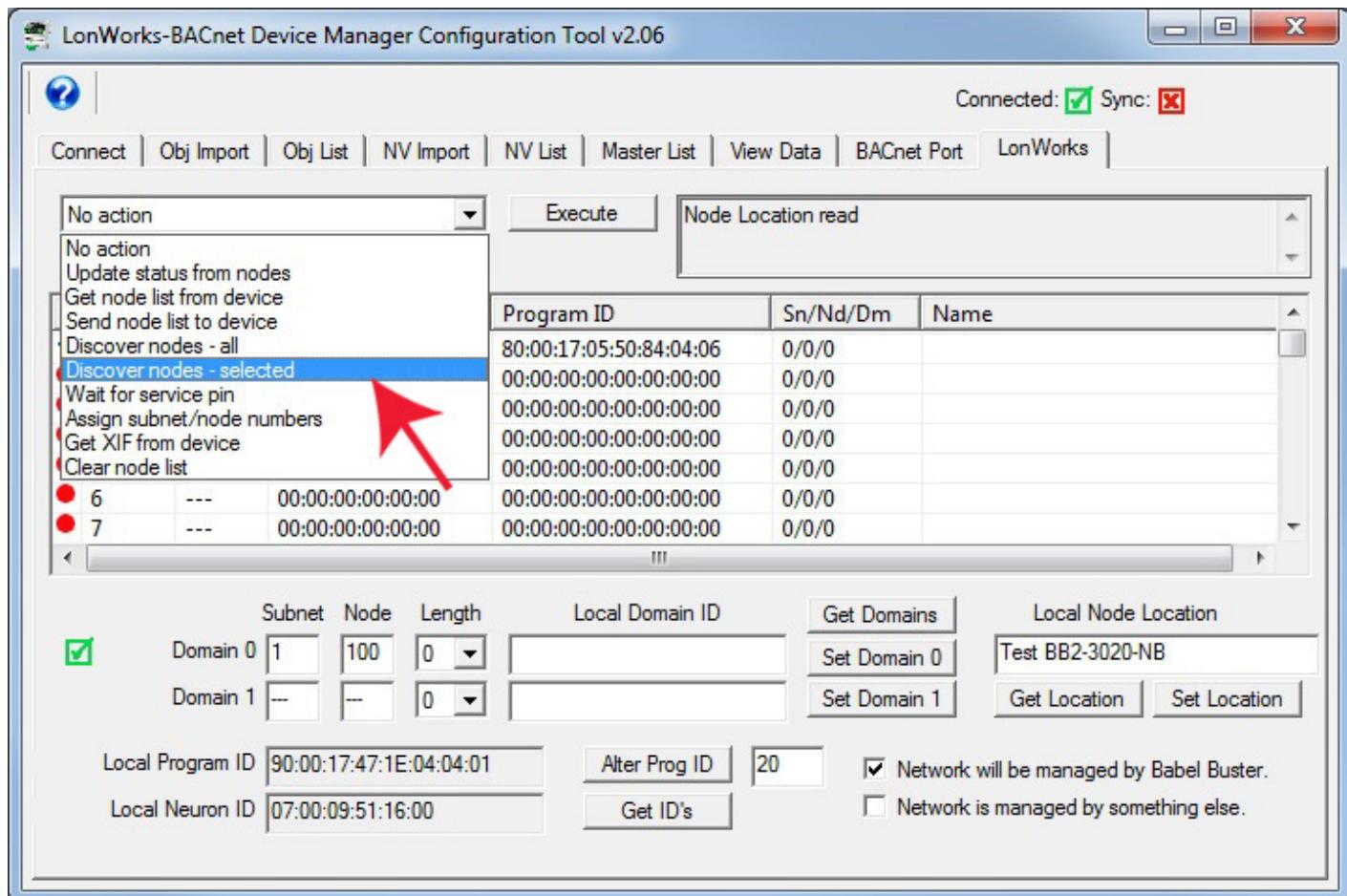
As with the other discovery methods, you also need to check 'Node Selected'. Then click Apply (not Add Node).



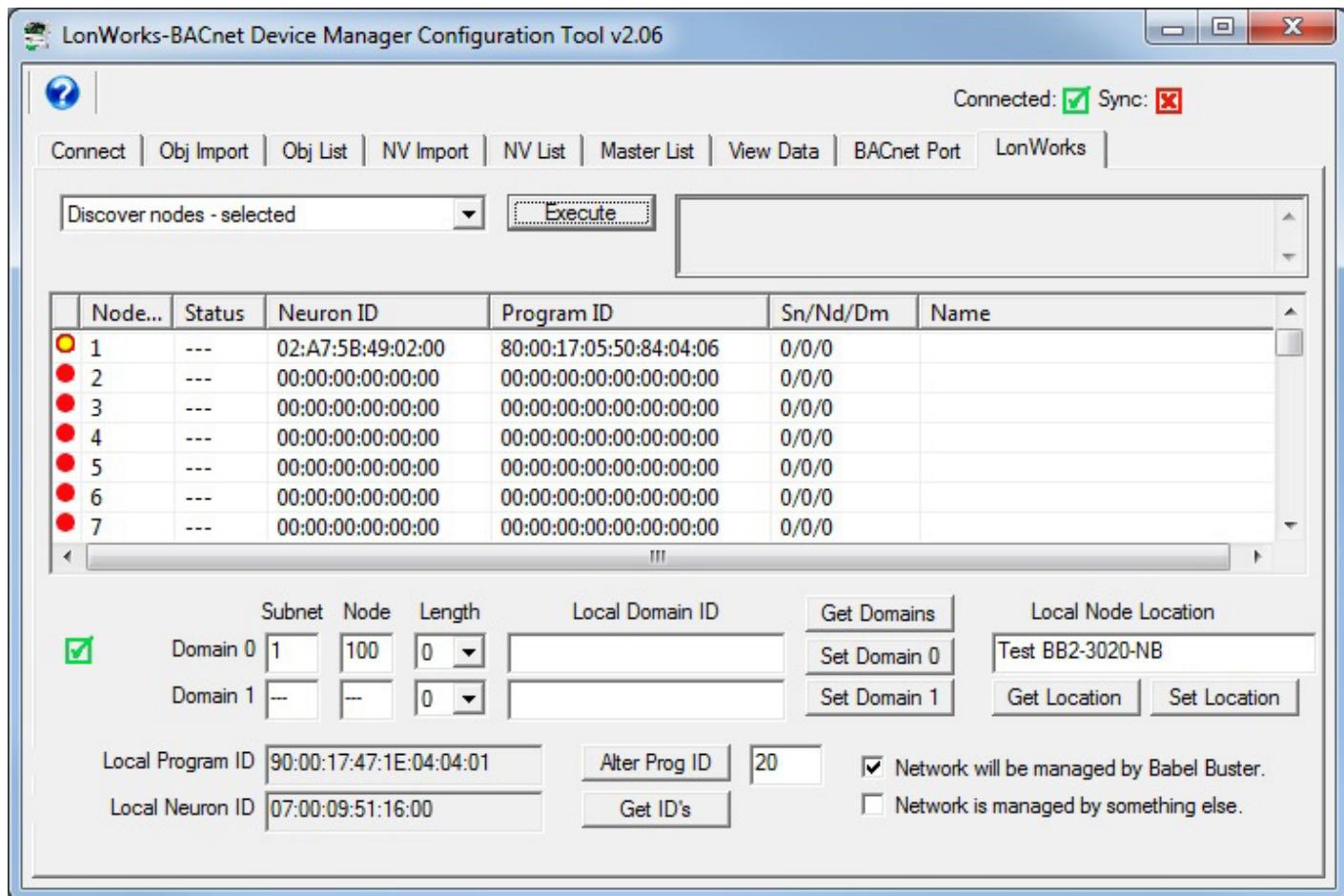
The node table now shows the check mark in the icon column as for other discover methods above, but now also shows a Program ID.



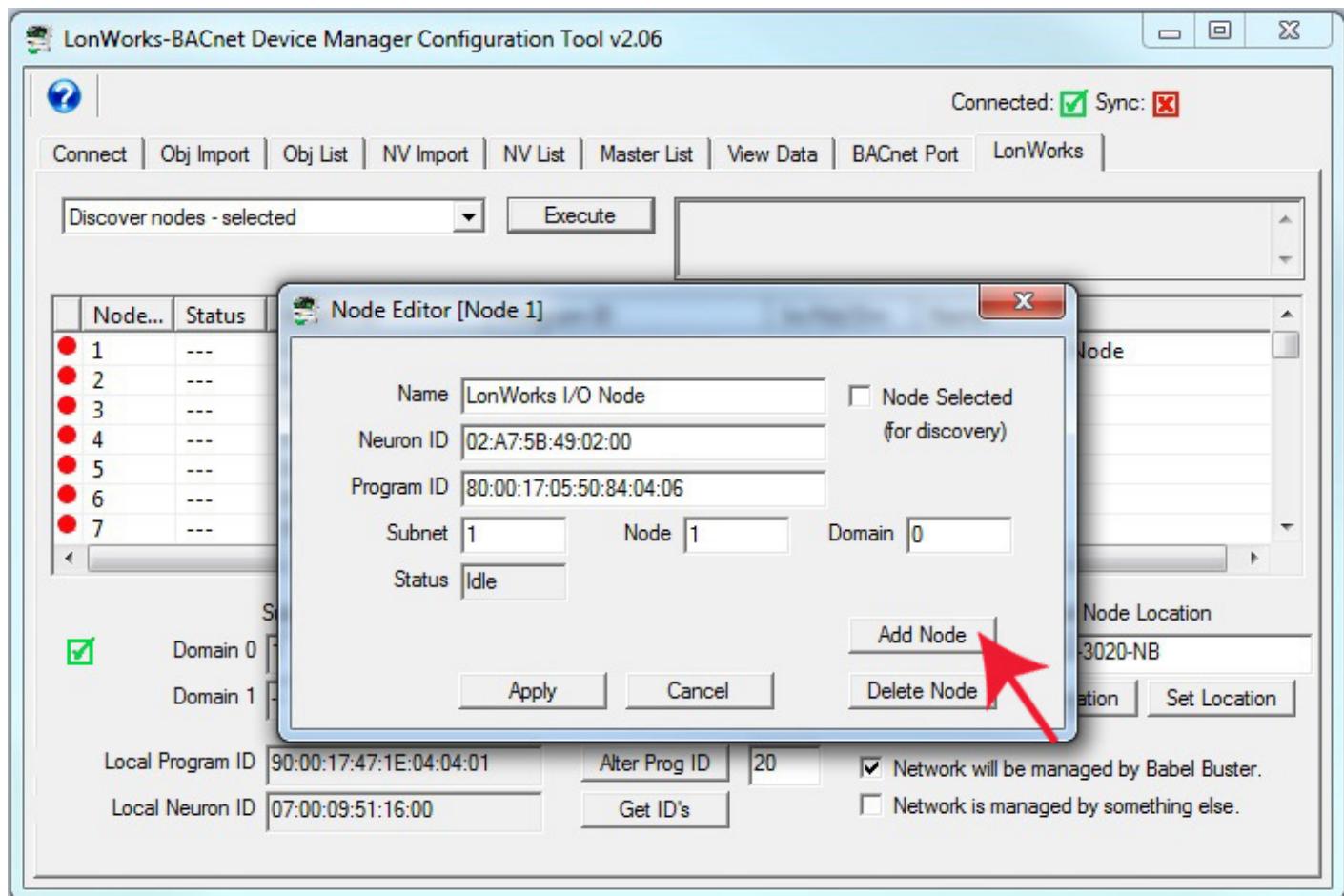
Next, select 'Discover nodes - selected' from the list and click Execute. Note that you can discover more than one node using this method. Simply use the Node Editor to enter the same Program ID in multiple table entries, and check 'Node Selected' for each.



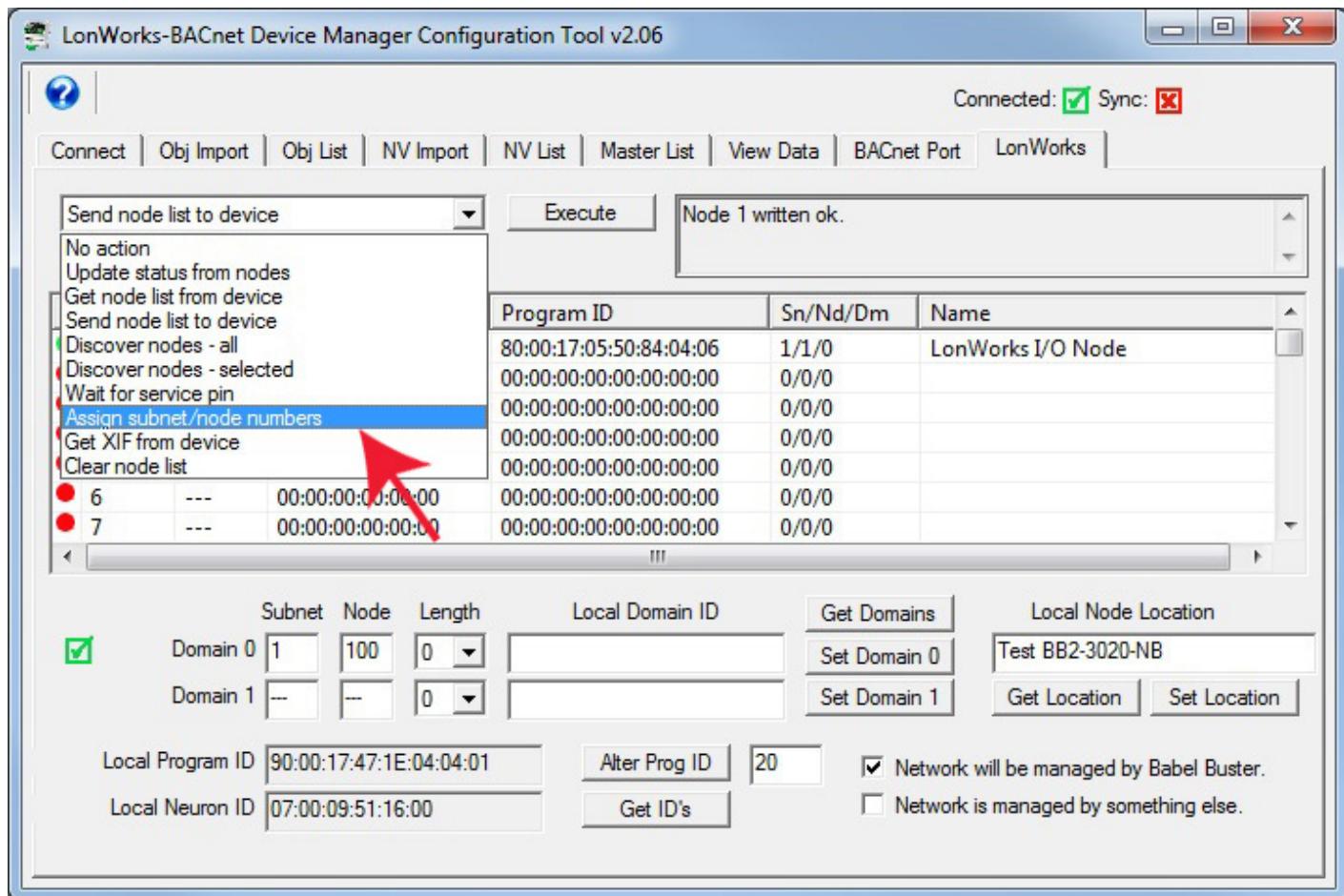
Refer to the discussion above about discovering all nodes if you're wondering what is going on at this point regarding how it discovers nodes. The only difference here is that the configuration tool is simply discarding replies from any nodes that do not match the Program ID provided. When the node is discovered, its icon will change to yellow with a red circle.



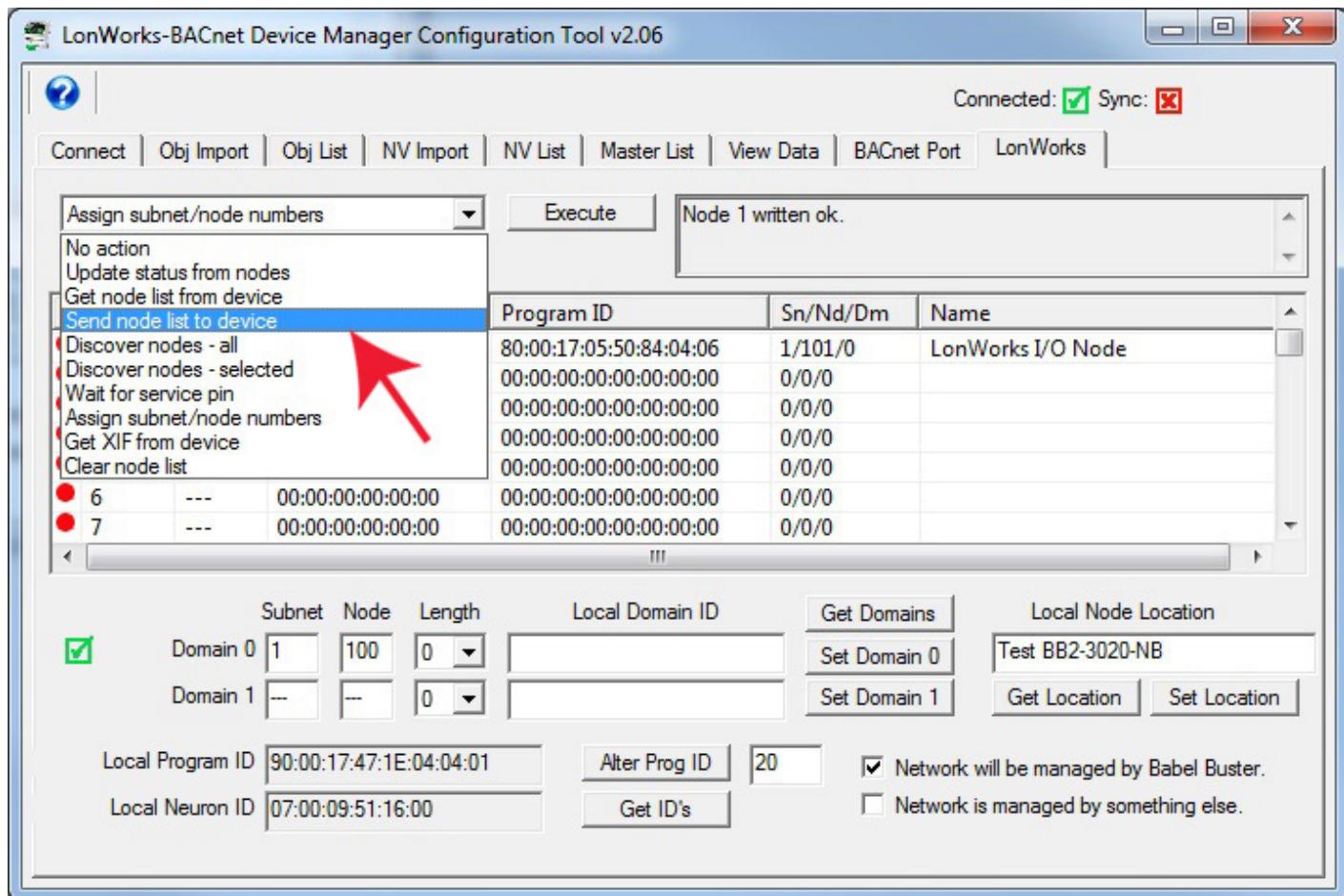
At this point, double click the node table entry to open the Node Editor. Provide a name and subnet/node if desired (see comments about this under discussion of all nodes above). Click Add Node.



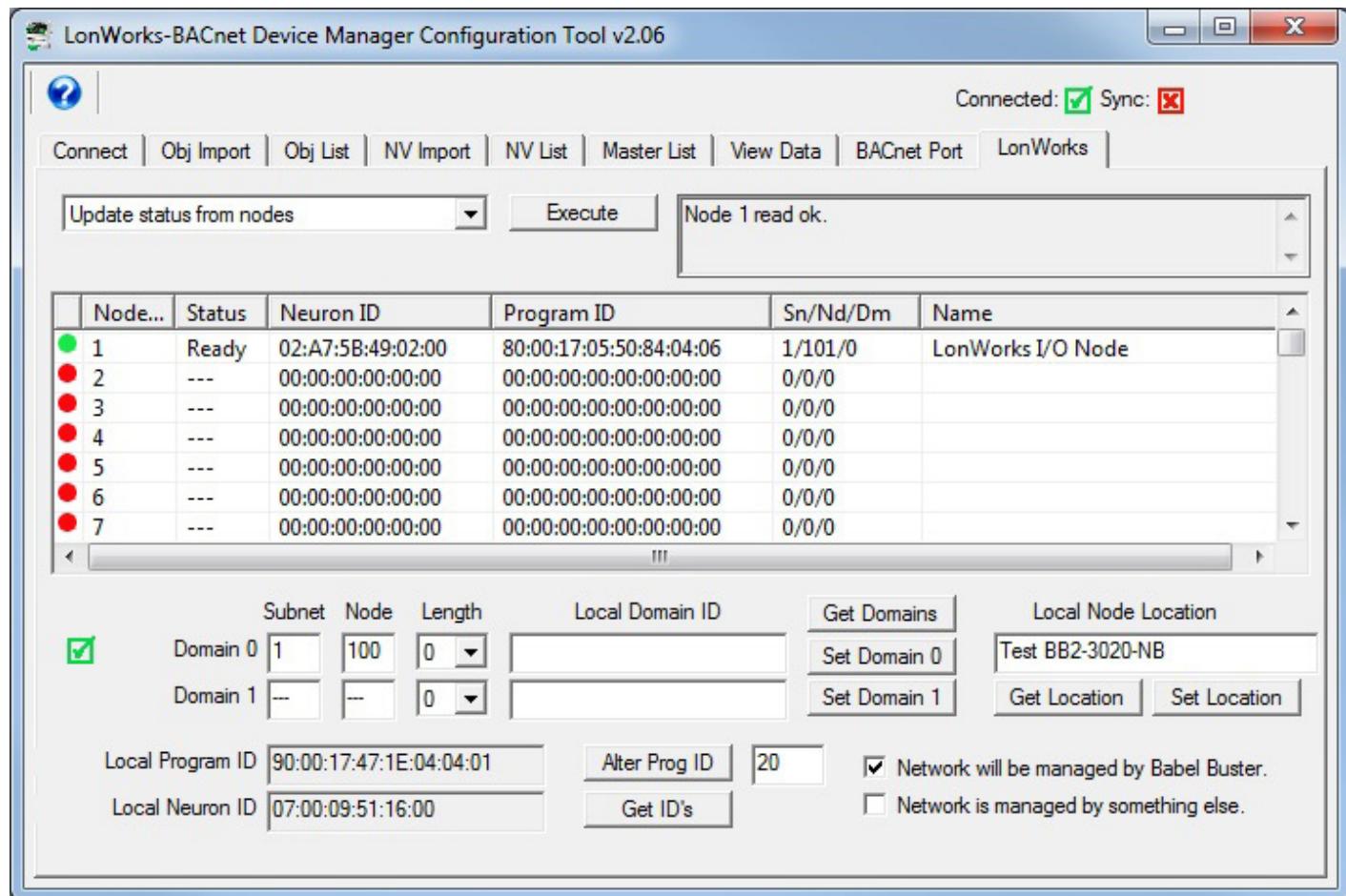
Most often, you will want to let the gateway assign suitable subnet/node numbers.



Send the node table to the gateway device as for the other methods above. Until you do this, the node is only known to the configuration tool. You need to populate the table in the device before the gateway will start to communicate with it.



Next, select 'Update status from nodes' and click Execute to confirm that this newly discovered node is now communicating successfully. At this point, you are ready to send the rest of the configuration from the NV List and Master List to the device. Before sending either of those lists, be sure to go to the BACnet Port page and send the BACnet device information to the gateway. The object counts are especially important for completing configuration successfully.





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
pcox <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

nf <n> - fetches whatever NV is defined at NV table map entry #n. View data using **nvs <n>**.
nu <n> <x1 ... xN> - updates whatever NV is defined at NV table map entry #n using hex bytes x1..xN.

nvx <n> - display as XML the configuration found in NV table map entry #n.
ndevx <n> - displays node configuration information, 0=local/self, 1..max=remote nodes, XML format

nvs <n> - display status, NV selector, and any pending diagnostic data for NV at table entry #n.
Display will be "S[] Sel=... Data: ..."

Sel= will be followed by 4-digit hex selector if it has been retrieved from the node
Data: will be followed by the data from the last "nf" command, cleared after displayed
Contents of S[] may be:

T - if tag timeout or message failed completion

N - if device not ready
E - if API returned an error code
F - if NV fetch is pending
U - if NV update is pending

nds <n> - shows node status for node table entry #n.

Display will be " [y] N: ... P ..." where Neuron ID follows N, program ID follows P.

Contents of [] may be:

S - if node has Neuron ID
D - set domain successful
Q - query domain successful
R - if node is considered 'ready'
W - if domain query came back with wrong subnet/node
F - failed domain set
N - failed domain query
E - if API reported an error

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

If the gateway is generally working but specific variables or objects are not correct, check the Reliability property of the BACnet object in question. If non-zero, it is telling you what the problem is. Refer to Appendix C for a list of reliability codes specific to the gateway, and their meaning.

The LED indicators on the gateway device provide certain clues when things are not working right.



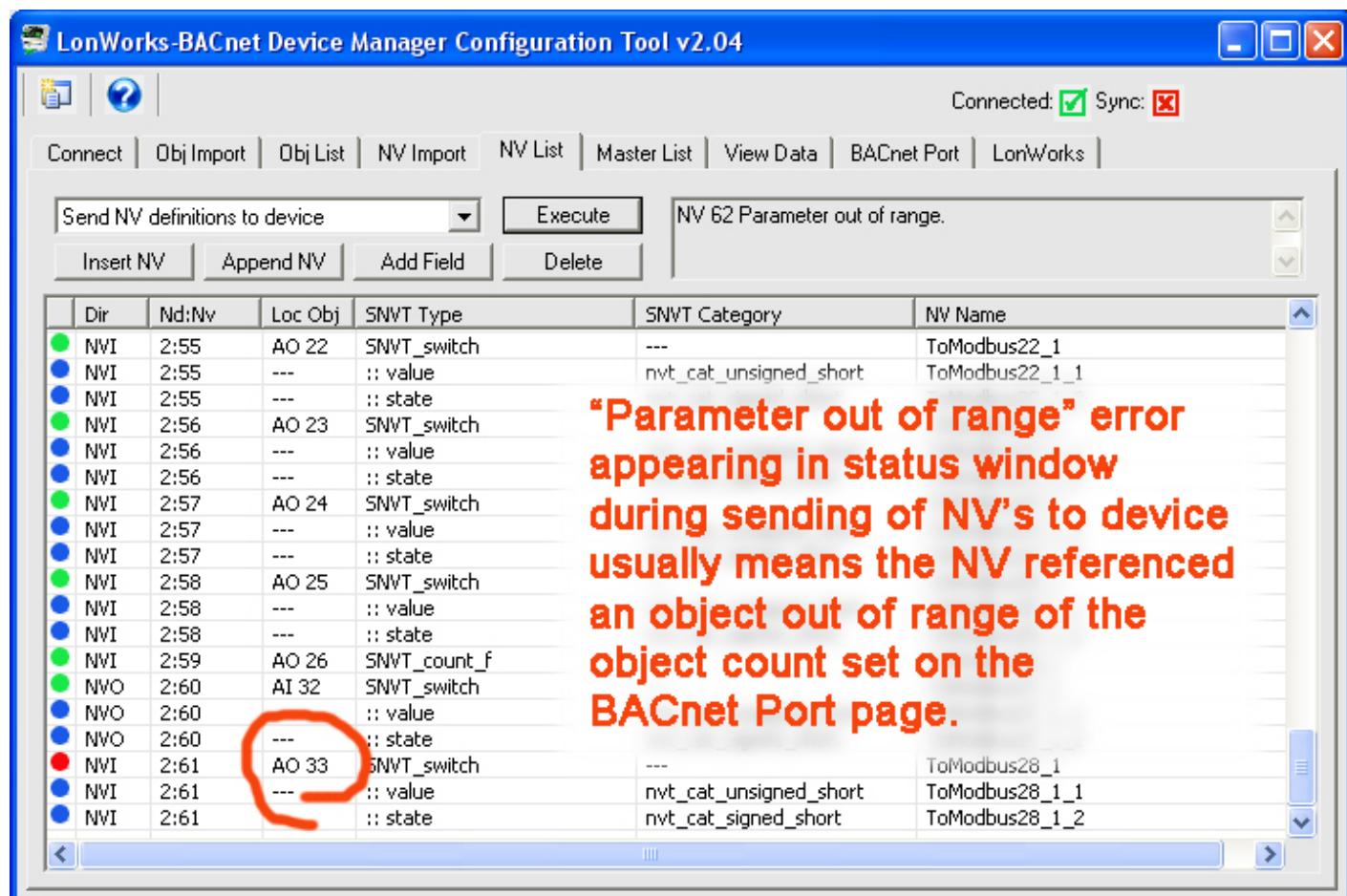
The LonWorks indicator LEDs display LonWorks network activity. Note, however, that activity of these LEDs is not only affected by configuration of the device, but by whether the device is commissioned on the network and whether other devices are also communicating.

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, or node management message was sent.	Red solid on indicates Neuron chip is not running. Brief flash of red indicates error in processing NV request or node management request.
	Green flash indicates NV fetch response was received by gateway or other action completed successfully.	Green indicates gateway's host processor is communicating with LonWorks Neuron chip.

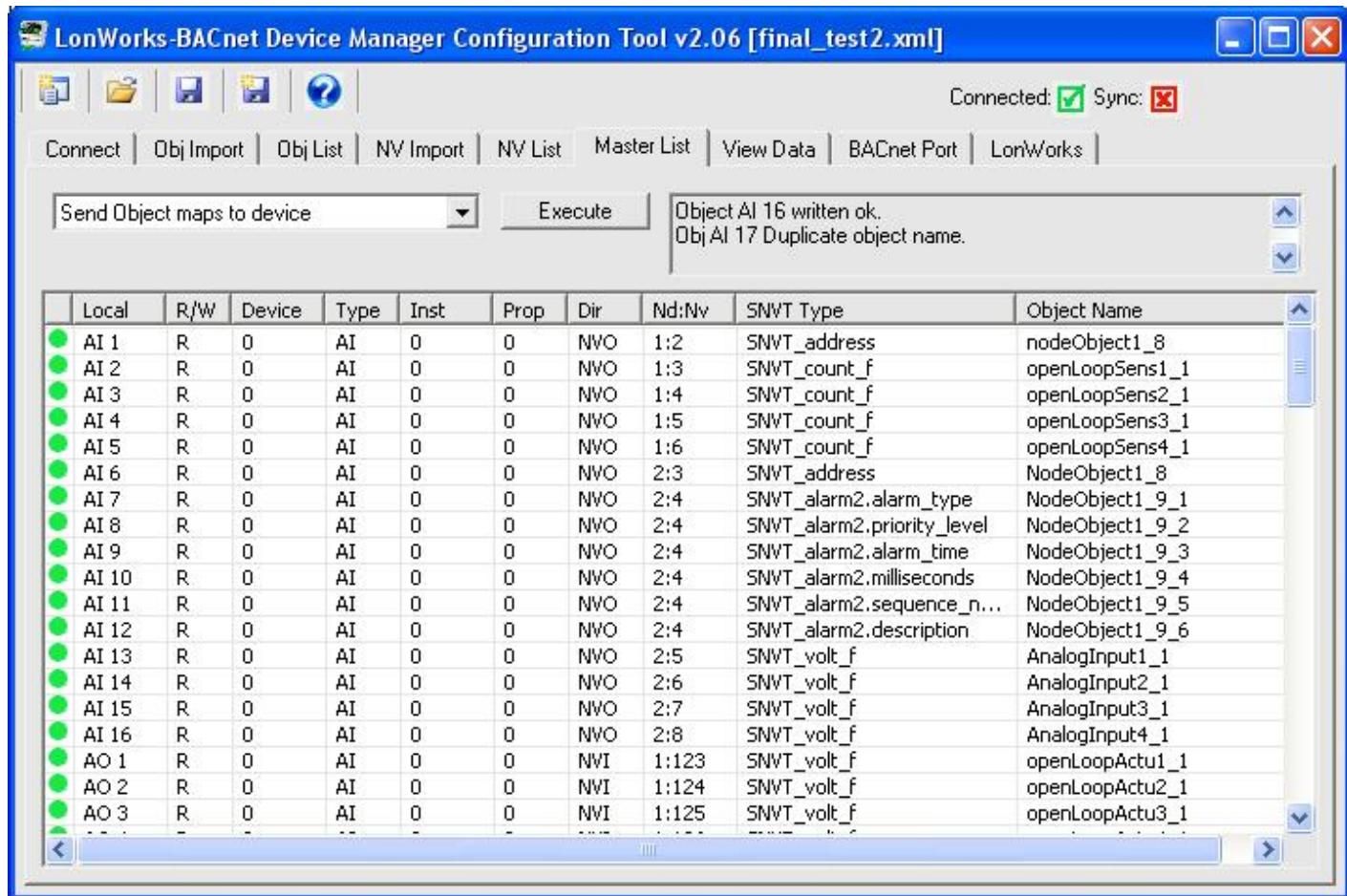
The "wink" behavior is invoked by sending a wink command to the BB2 gateway via the LonWorks network. This is generally just a diagnostic to see if you are successfully communicating with the device via LonWorks. Other than the few seconds it takes to execute the wink, the device will always be in "normal"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

B.2 Errors When Sending Configuration

The message "Parameter out of range" when sending Network Variables to the device can mean any of several things; however, the configuration tool will attempt to block many of the possibilities. One common error that can/will occur is referencing a BACnet object that is not within the range of BACnet object counts set on the BACnet Port page. If you did not "Recalculate" and reconfigure the object counts in the Babel Buster device, then this error is likely to happen.

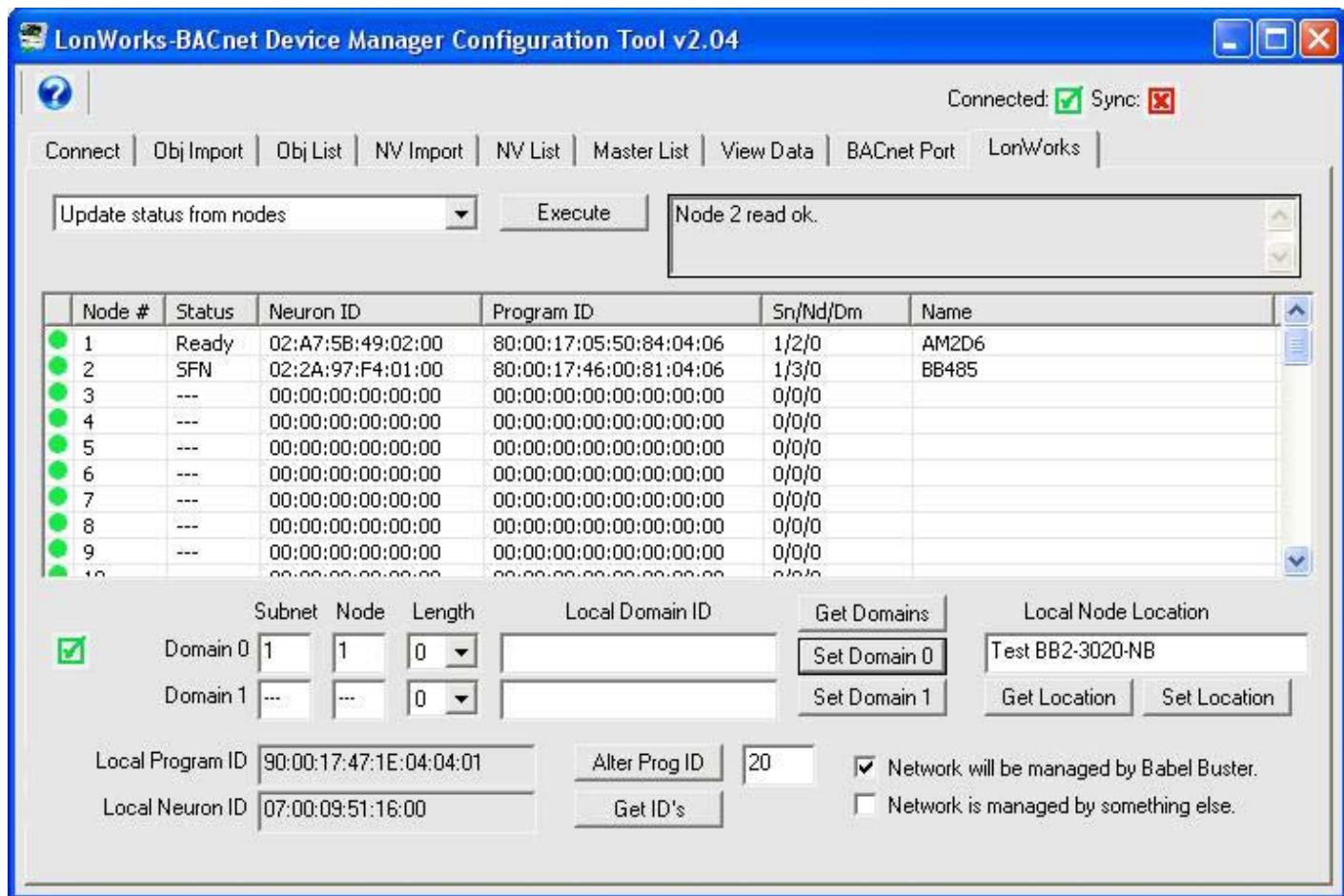


BACnet protocol does not allow any two objects in the same device to have the same object name. If you are editing object names in the configuration tool, the tool will check for duplicate names and block your attempt. However, if you have imported any part of the configuration, bypassing the name editor, then it is still possible to end up with duplicate names present in the tool. You will not discover this problem until you attempt to send the duplicate names to the Babel Buster gateway device.



B.3 Node Status Not "Ready"

The following screen shot shows what you will typically see if a node on the network (in this example, node #2) is not "Ready" for the Babel Buster gateway to poll its network variables:



The above instance shows that node #2 has reported its Neuron ID and Program ID. The configuration tool wants to set this node to subnet 1, node number 3. However, it has not been successful as indicated by the "SFN" status.

The above instance occurred when the device at node #2 in the table happened to already be assigned to the same subnet and node number as the Babel Buster gateway itself. As a result, responses from node #2 will not be received. In fact, they will not be sent by node #2 because it thinks it should be sending the response to subnet 1, node 1, which is its own address, and it will therefore discard the transmission to itself.

If you happened to have a LonWorks network interface and the Nodeutil program handy, connecting to node #2 (from table above) and checking its domain table will provide the results illustrated here:

C:\Keil\BB2-3020-NB_v3.02.3\NodeUtil.exe

```

Transaction timeouts          = 0
Receive trans full errors   = 0
Lost msgs <no app buff>     = 0
Missed msgs <no net buff>   = 0
Packets received by device  = 41
Packets addressed to device = 15
Messages sent to MAC layer   = 16
Retries                      = 0
Backlog overflows           = 0
Late acks or responses      = 1
Collisions detected         = 0
EEPROM lock                  = Clear
Last reset cause             = Power-up
Device state                 = Configured, On-line
Firmware version number      = 15
Build number                 = 0
Neuron model                = 3150L
Last error logged            = None
Do you want to clear node status? <Y/[N]>:N
DEVICE:1> Device <D>omain table
Enter domain table index <0-1> [All] : 
Index  Size  Subnet  Node  Auth Key  Domn ID
0      0      1        1      FF FF FF FF FF FF
1      0      1        *1    FF FF FF FF FF FF
DEVICE:1>

```

The resolution in this case is to simply change the node number of the Babel Buster gateway to something else as illustrated below.

LonWorks-BACnet Device Manager Configuration Tool v2.04

Connected: Sync:

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

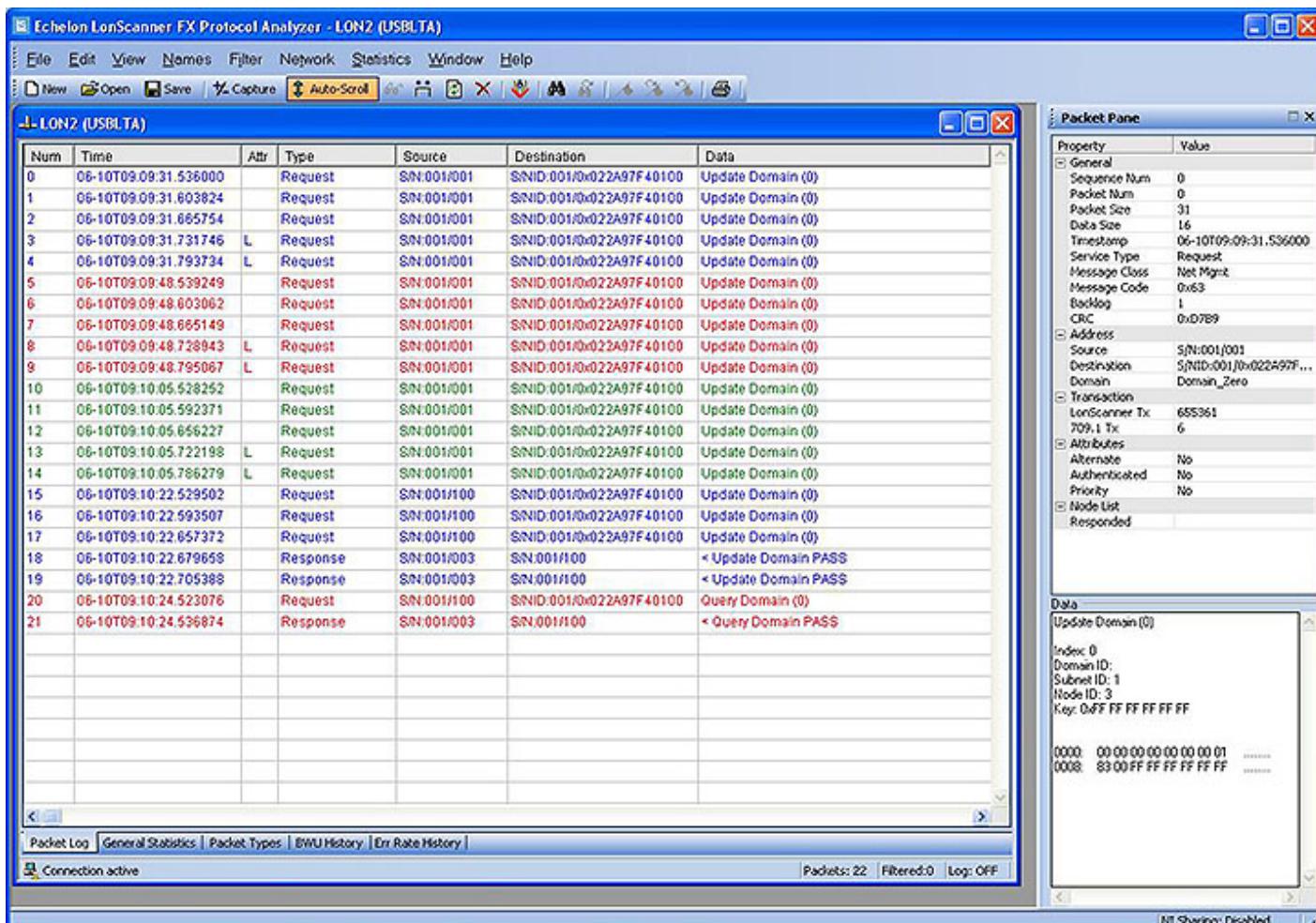
Update status from nodes Domain 0 written

Node #	Status	Neuron ID	Program ID	Sn/Nd/Dm	Name
1	Ready	02:A7:5B:49:02:00	80:00:17:05:50:84:04:06	1/2/0	AM2D6
2	SFN	02:2A:97:F4:01:00	80:00:17:46:00:81:04:06	1/3/0	BB485
3	---	00:00:00:00:00:00	00:00:00:00:00:00:00:00	0/0/0	
4	---	00:00:00:00:00:00	00:00:00:00:00:00:00:00	0/0/0	
5	---	00:00:00:00:00:00	00:00:00:00:00:00:00:00	0/0/0	
6	---	00:00:00:00:00:00	00:00:00:00:00:00:00:00	0/0/0	
7	---	00:00:00:00:00:00	00:00:00:00:00:00:00:00	0/0/0	
8	---	00:00:00:00:00:00	00:00:00:00:00:00:00:00	0/0/0	
9	---	00:00:00:00:00:00	00:00:00:00:00:00:00:00	0/0/0	
10	---	00:00:00:00:00:00	00:00:00:00:00:00:00:00	0/0/0	

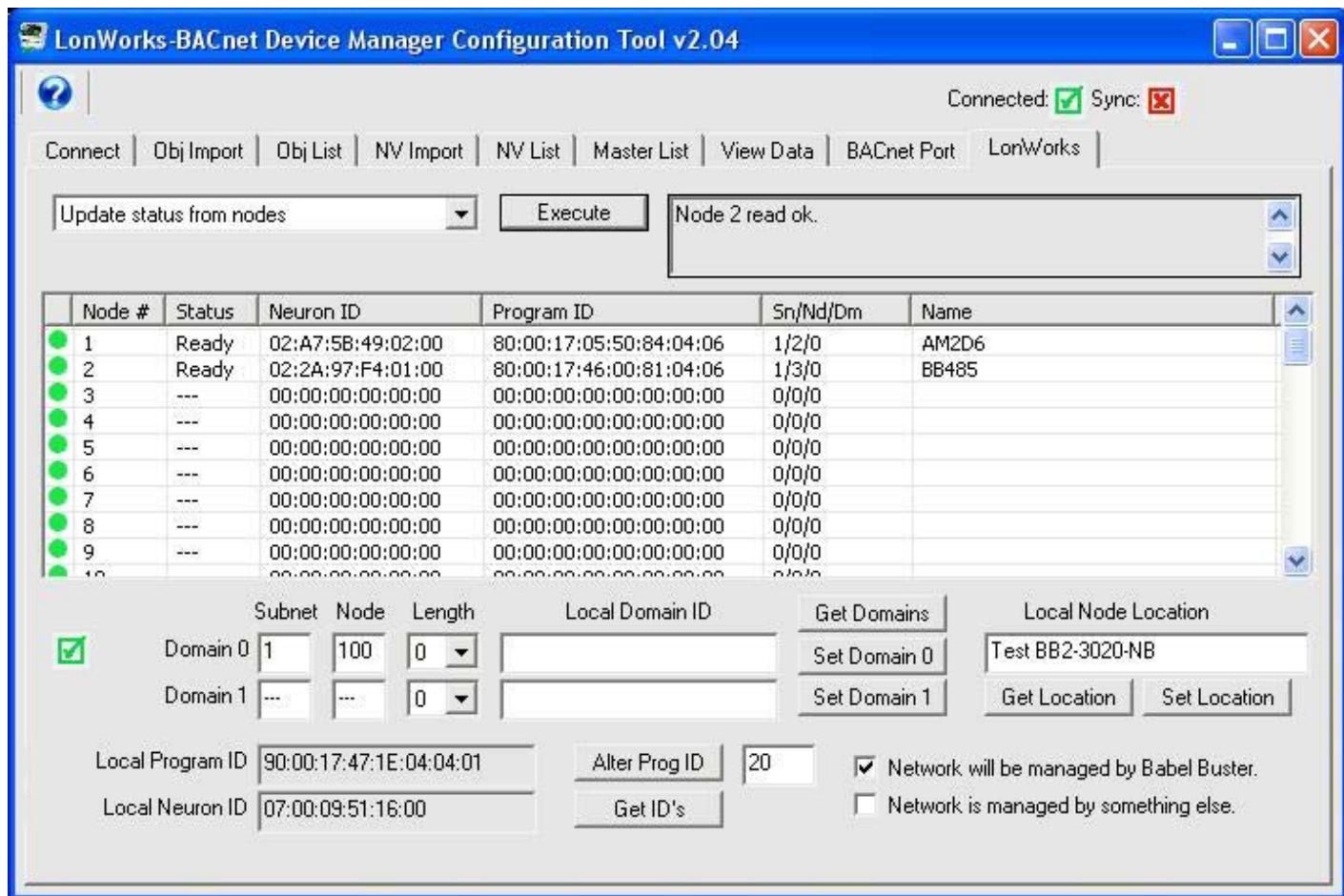
Subnet Node Length Local Domain ID Get Domains Local Node Location
 Domain 0 1 100 0 Test BB2-3020-NB
 Domain 1 --- 0

Local Program ID 90:00:17:47:1E:04:04:01 Alter Prog ID 20 Network will be managed by Babel Buster.
Local Neuron ID 07:00:09:51:16:00 Network is managed by something else.

The activity on the network during this exercise using LonScanner is illustrated here:



The LonScanner network caputure illustrates the non-communicating state through record 14, then the node number was changed, and communication is successful. The resulting screen shot of the LonWorks page is shown below.



Nobody expects you to be using Nodeutil and LonScanner to decipher this problem. It is illustrated here to demonstrate the problem and solution. Therefore, in practice, if the Babel Buster gateway is managing the network ("Network will be managed by Babel Buster") but you still cannot communicate with the node, then simply try changing the node number of the gateway itself. Do this by changing the node number on the LonWorks page, and then click Set Domain.

B.4 Cannot Discover Node

Node discovery requires being able to receive responses from devices on the same domain as the gateway. The Babel Buster gateway will broadcast a message telling all devices to "respond to query". Devices will report their identity. However, if the device is on a domain different than the gateway, the responses will not be received by the gateway.

You have two choices for resolving this matter: (a) Use other tools (e.g. Nodeutil or other network management tool) to find out what the domain of the device is; (b) Use the service pin method to locate the device and make it known to the gateway.

If the gateway is being used on a managed network, i.e., network managed by something other than Babel Buster, then your only choice is to find out (from the network management tool) what domain the devices are set to, and change the gateway's domain to match that domain. The service pin method will locate the device, but the gateway will still be unable to communicate because it has been instructed to not make any changes to device domain tables (implied in configuring the gateway for use on a network managed by something other than Babel Buster).



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-NB 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 while 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 your 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 that 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-NB 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.

C.2 BACnet Object Reliability Codes

The errors you are most likely to see in relation to the Babel Buster gateway polling LonWorks Network Variables, or in relation to the gateway polling BACnet objects in other devices, are classified as "manufacturer specific" codes, and these are as follows for the Babel Buster gateway.

The reliability code will provide an indication of problems if non-zero. The following reliability codes can occur when the Babel Buster is operating as a BACnet client (polling other BACnet devices) and there is a problem related to the attempt to access the remote BACnet device:

- 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.

The following reliability codes pertain to the LonWorks side, and indicate a problem attempting to communicate with the LonWorks device:

- 90 - NV Fail - most often incorrect NV index or incorrect NV size if writing to the LonWorks device
- 91 - Node Fail - unable to communicate with the node
- 92 - API Error - indicates a problem with the host processor's attempt to communicate with the internal LonWorks processor.

Note that "Node Fail" can have a multitude of causes. If a node had previously communicated successfully and then fails, it can mean something simple like disconnected network cable. However, if the node has never communicated, there is probably a fundamental configuration problem and checking node status on the LonWorks page will indicate that the node is not in the "Ready" state. Proceed to attempt to commission the device as discussed in the LonWorks section.



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.

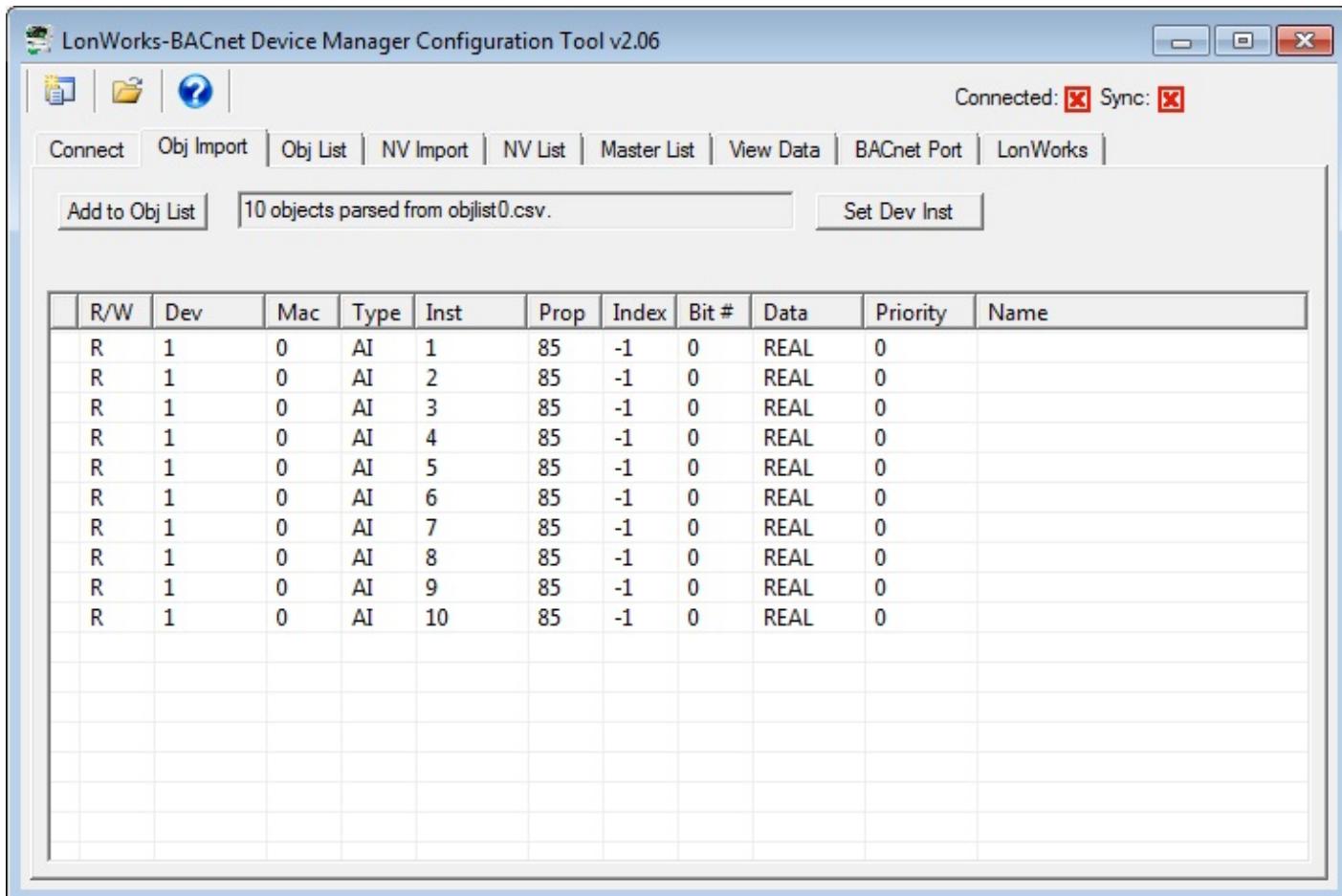
Label	Content	What it means
RW	Either R or W (or W+) (default is R)	Enter R to read the object, or W to write it. If omitted, defaults to R. Use W+ for write on Update.
DEV	Device instance, e.g. 7171 (default is 1)	Enter the device instance of the remote BACnet server (slave). The remote device must 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.

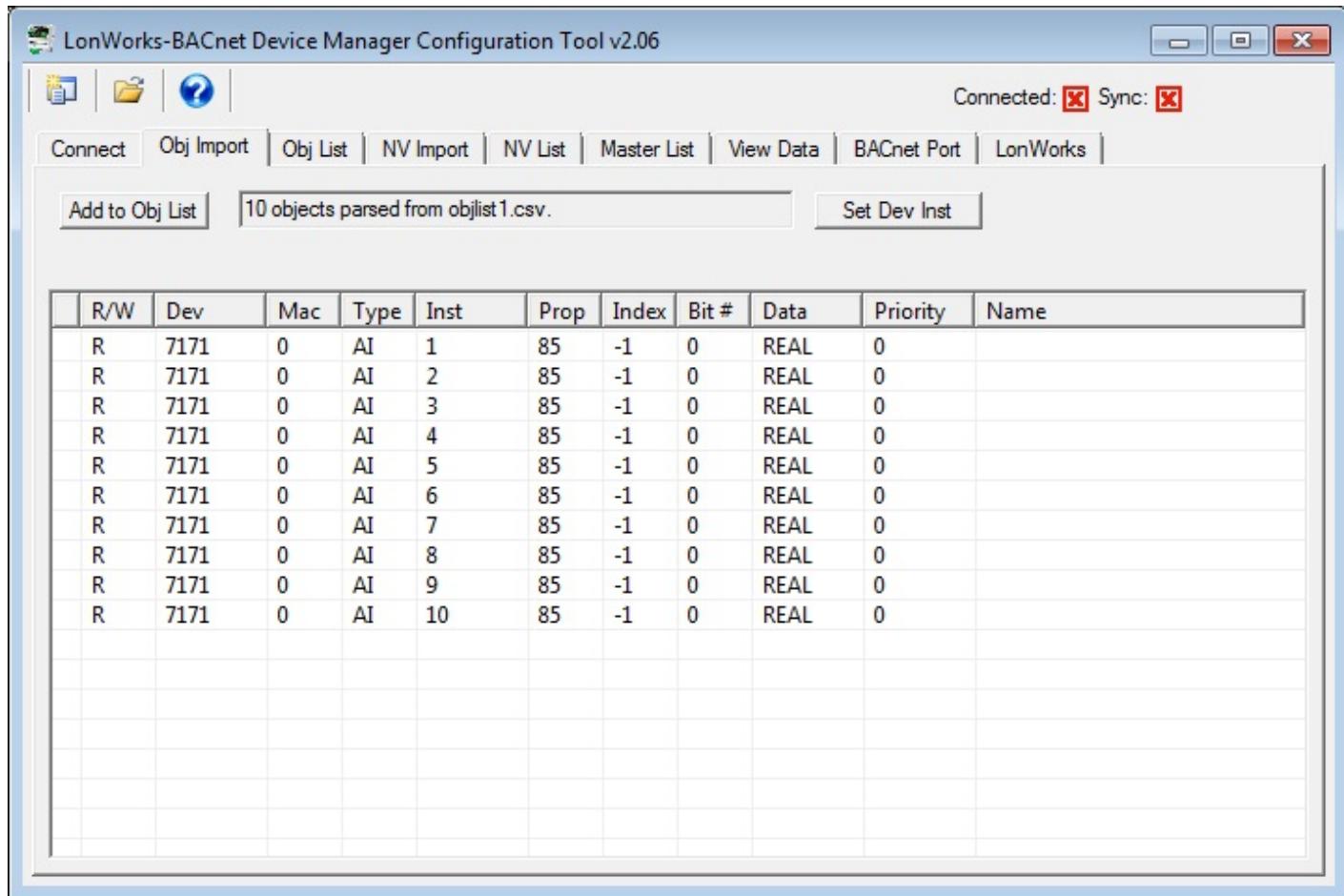
D.2 Example CSV Files and Imports

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.
--	--



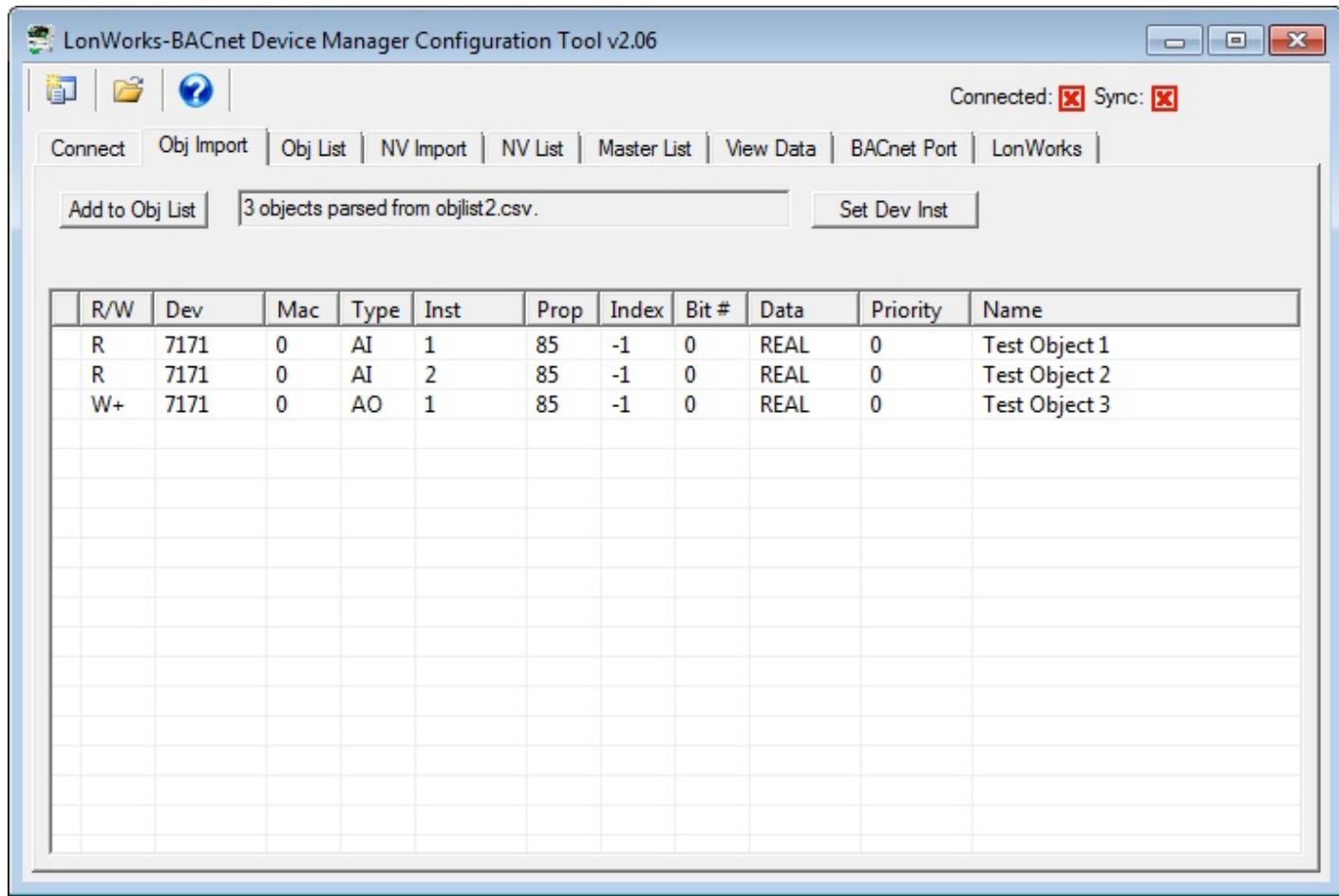
DEV,OBJTYPE,OBJINST 7171,AI,1 7171,AI,2 7171,AI,3 7171,AI,4 7171,AI,5 7171,AI,6 7171,AI,7 7171,AI,8 7171,AI,9 7171,AI,10	A minor expansion of the above example adds the device instance column. The following screen shot is the result of importing this CSV file.
--	--



```
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.

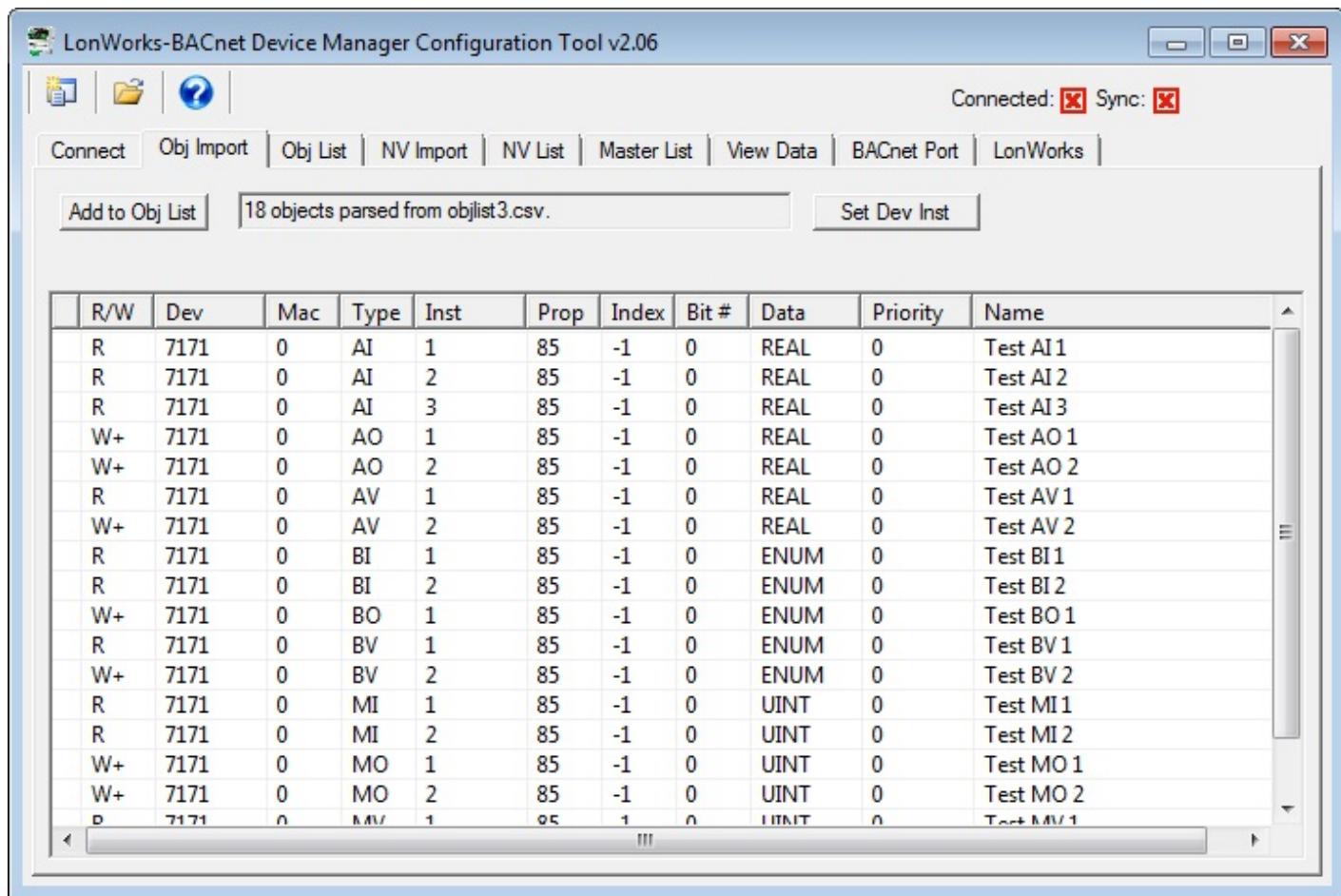
Importing this CSV results in the following screen shot.



```
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.



```

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.

LonWorks-BACnet Device Manager Configuration Tool v2.06

Connected: Sync:

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

Add to Obj List 18 objects parsed from objlist4.csv. Set Dev Inst

R/W	Dev	Mac	Type	Inst	Prop	Index	Bit #	Data	Priority	Name
R	7171	0	AI	1	85	-1	0	REAL	0	Test AI1
R	7171	0	AI	2	85	-1	0	REAL	0	Test AI2
R	7171	0	AI	3	85	-1	0	REAL	0	Test AI3
W+	7171	0	AO	1	85	-1	0	REAL	0	Test AO1
W+	7171	0	AO	2	85	-1	0	REAL	0	Test AO 2
R	7171	0	AV	1	85	-1	0	REAL	0	Test AV1
W+	7171	0	AV	2	85	-1	0	REAL	0	Test AV 2
R	7171	0	BI	1	85	-1	0	ENUM	0	Test BI1
R	7171	0	BI	2	85	-1	0	ENUM	0	Test BI 2
W+	7171	0	BO	1	85	-1	0	ENUM	0	Test BO1
R	7171	0	BV	1	85	-1	0	ENUM	0	Test BV1
W+	7171	0	BV	2	85	-1	0	ENUM	0	Test BV 2
R	7171	0	MI	1	85	-1	0	UINT	0	Test MI1
R	7171	0	MI	2	85	-1	0	UINT	0	Test MI 2
W+	7171	0	MO	1	85	-1	0	UINT	0	Test MO1
W+	7171	0	MO	2	85	-1	0	UINT	0	Test MO 2
D	7171	0	MV	1	85	-1	0	UINT	0	Test MV1

LonWorks-BACnet Device Manager Configuration Tool v2.06

Connected: Sync:

Connect Obj Import Insert Obj

Object Editor

Object Type/Inst: Analog Input 1 Change

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:

Apply Cancel

Nd:Nv	Lo
1:1	AI
1:2	AI
1:3	AI
1:4	AO
1:5	AO
1:6	AI
1:7	AO
1:8	AI
1:9	AI
1:10	AO
1:11	AI
1:12	AO
1:13	AI
1:14	AI
1:15	AO
1:16	AO
1:17	AI



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_PROPORIONAL_CONSTANT = 93,
PROP_PROPORIONAL_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_UTCTOFFSET = 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,
PROPUSES_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 CSV Network Variable List Format

F.1 Data Labels on Header Line

The required format for importing LonWorks network variable 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. Unlike the BACnet object CSV import where most columns are optional, the NV import requires all but the name in order to be functional. 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.

Label	Content	What it means
NODE	1..50	This number will correspond to the node number in the gateway's node table. The node table will contain addressing information such as the target device's Neuron ID, subnet and node numbers, and domain table index.
NVINDEX	0..16383	The network variable index identifies the network variable in the target device. You would obtain this information from the manufacturer of the device, or from the XIF file for that device.
DIR	1=NVI at target, 0=NVO at target	Identifies whether the network variable of interest is an NVI (network variable input) or NVO (network variable output) at the target device. The gateway can read from an NVO, and optionally write to an NVI. The gateway cannot write to an NVO in the target device.
SERVICE	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.
SNVT	SNVT index 1..177 (zero if UNVT)	Enter the LonMark SNVT Index if the NV is a standard LonMark variable type. If it is manufacturer defined (UNVT), enter zero here.

UNVTSIZE	UNVT size 1..31 (zero if SNVT is valid non-zero index)	If SNVT is zero, then a non-zero number from 1 to 31 must be entered here to tell the gateway what size variable to expect.
NAME	Any unique character string up to 20 characters	Enter an alphanumeric name for the NV map. This name is for reference only, and does not affect the names of BACnet objects.

F.2 Example CSV File and Import

The following example illustrates the CSV file format and resulting screen shot following import.

NODE,NVINDEX,DIR,SERVICE,SNVT,UNVTSIZE,NAME 1,5,0,R,66,0,AnalogInput1_1 1,6,0,R,66,0,AnalogInput2_1 1,7,0,R,66,0,AnalogInput3_1 1,8,0,R,66,0,AnalogInput4_1 1,9,1,W+,95,0,DiscreteOutp1_1 1,10,1,W+,95,0,DiscreteOutp2_1 2,5,0,R,66,0,AnalogInput1_1 2,6,0,R,66,0,AnalogInput2_1 2,7,0,R,66,0,AnalogInput3_1 2,8,0,R,66,0,AnalogInput4_1 2,9,1,W+,95,0,DiscreteOutp1_1 2,10,1,W+,95,0,DiscreteOutp2_1	Importing this CSV, reflecting network variables in two different LonWorks nodes, results in the following screen shot.
---	---

Dir	Nd:Nv	Loc ...	SNVT Type	SNVT Category	NV Name
● NVO	1:5	---	SNVT_volt_f	---	AnalogInput1_1
● NVO	1:6	---	SNVT_volt_f	---	AnalogInput2_1
● NVO	1:7	---	SNVT_volt_f	---	AnalogInput3_1
● NVO	1:8	---	SNVT_volt_f	---	AnalogInput4_1
● NVI	1:9	---	SNVT_switch	---	DiscreteOutp1_1
● NVI	1:9	---	:: value	nvt_cat_unsigned_short	DiscreteOutp1_1_1
● NVI	1:9	---	:: state	nvt_cat_signed_short	DiscreteOutp1_1_2
● NVI	1:10	---	SNVT_switch	---	DiscreteOutp2_1
● NVI	1:10	---	:: value	nvt_cat_unsigned_short	DiscreteOutp2_1_1
● NVI	1:10	---	:: state	nvt_cat_signed_short	DiscreteOutp2_1_2
● NVO	2:5	---	SNVT_volt_f	---	AnalogInput1_1
● NVO	2:6	---	SNVT_volt_f	---	AnalogInput2_1
● NVO	2:7	---	SNVT_volt_f	---	AnalogInput3_1
● NVO	2:8	---	SNVT_volt_f	---	AnalogInput4_1
● NVI	2:9	---	SNVT_switch	---	DiscreteOutp1_1
● NVI	2:9	---	:: value	nvt_cat_unsigned_short	DiscreteOutp1_1_1
● NVI	2:9	---	:: state	nvt_cat_signed_short	DiscreteOutp1_1_2



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

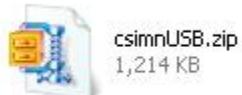
H.1 Driver Installation

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".



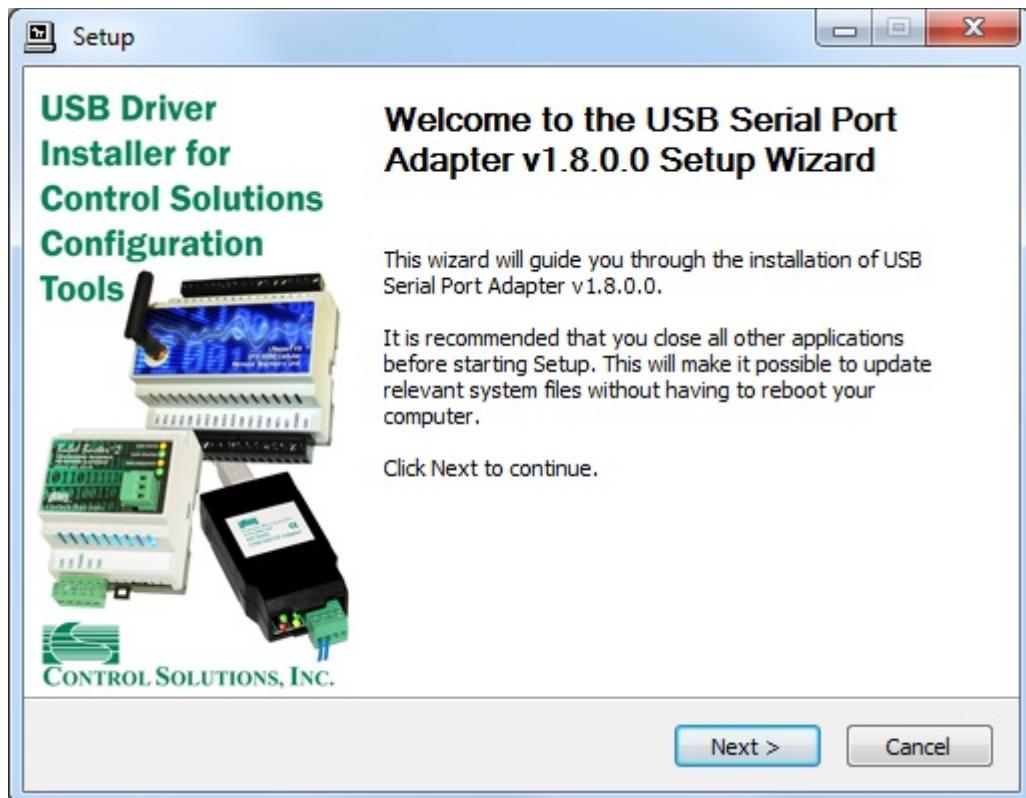
Unzip the contents of this file into a directory somewhere on your PC. The contents will look like this:

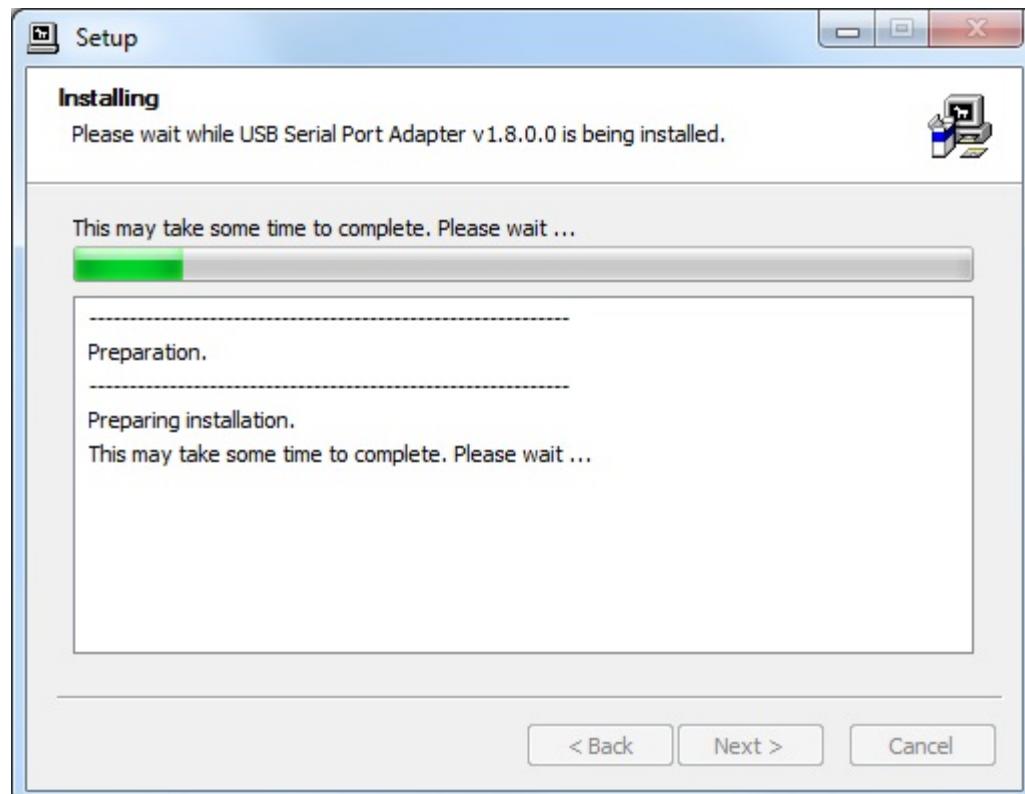
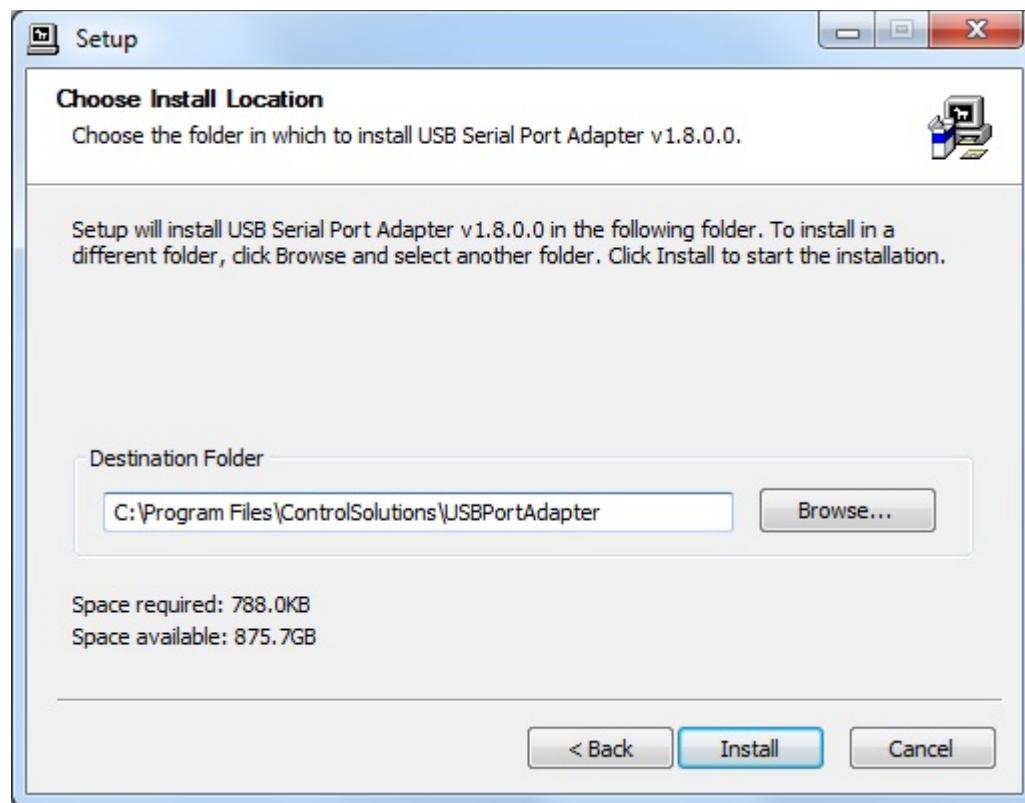


Double click "setup.exe". Say "yes" to any questions about whether to trust this software. Also, for Windows 8, you should right click on the setup file and "Run as administrator" - you will need to be logged in with administrator privileges.

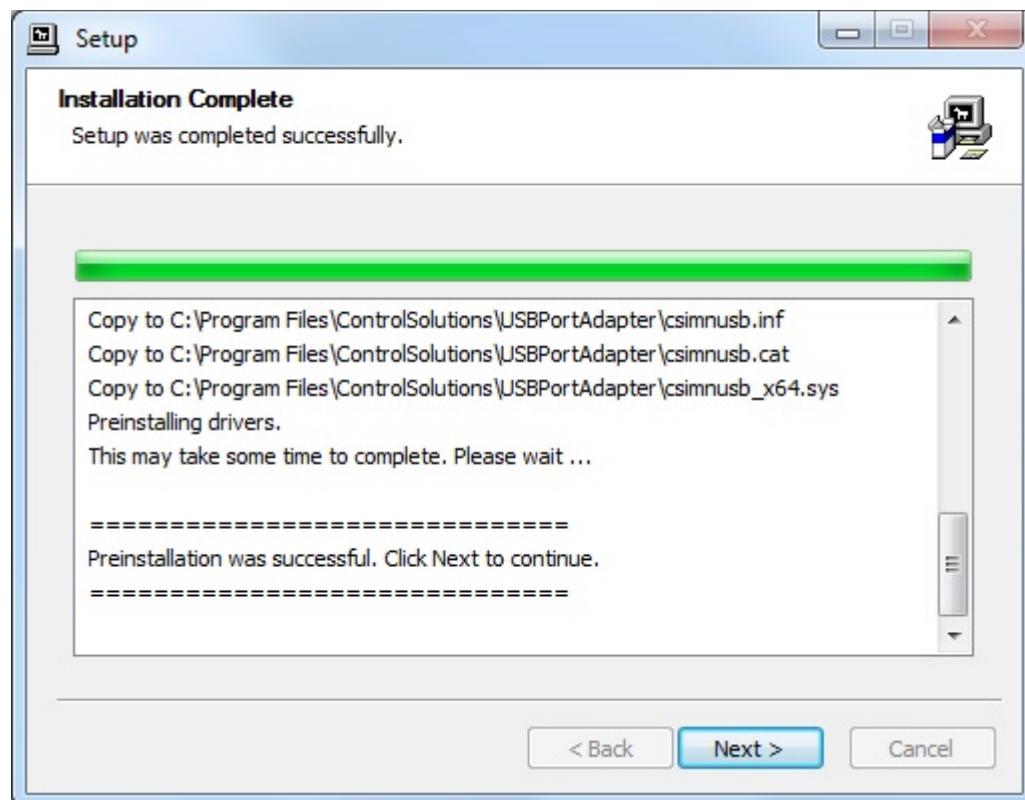
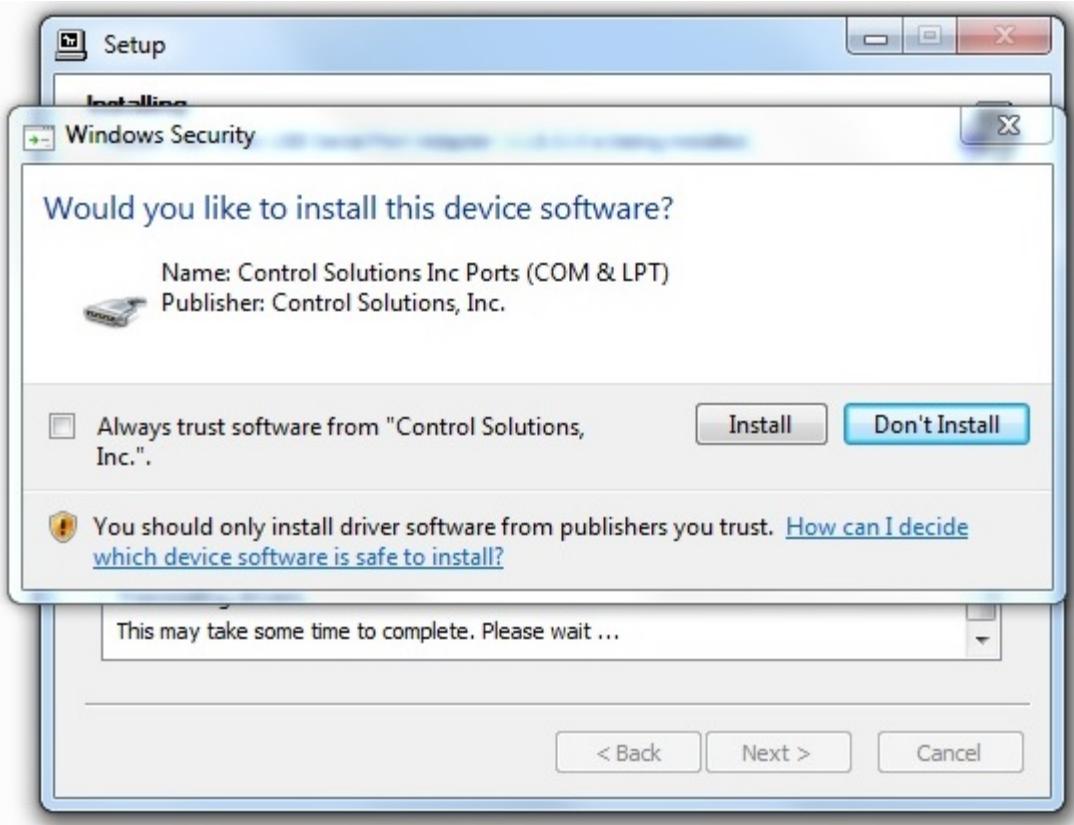
A sample of the series of screens you will see appears below. Basically all you need to do is follow the prompts, and click "yes", "next", "continue anyway", etc, as applicable.

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.)





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



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



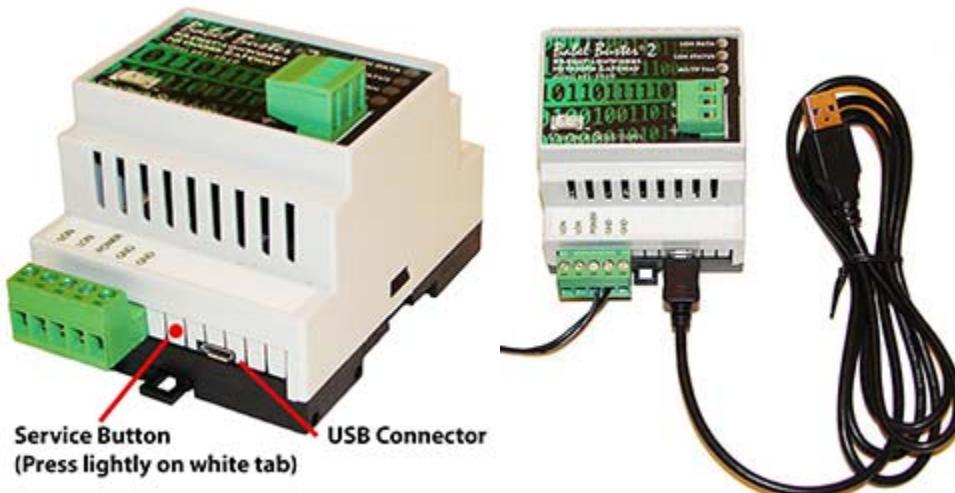


Appendix J Hardware Details

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-NB 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-NB IP gateway: Will behave the same as MS/TP, except the IP request/reply behavior will be delayed by several seconds after power-up indicatn on the LonWorks LEDs.



**LonWorks
Data &
Status LED
Indicators**



**LonWorks
Data &
Status LED
Indicators**

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	<p>Yellow flash indicates NV update was sent by gateway, or node management message was sent.</p> <p>Green flash indicates NV fetch response was received by gateway or other action completed successfully.</p>	<p>Red solid on indicates Neuron chip is not running. Brief flash of red indicates error in processing NV request or node management request.</p> <p>Green indicates gateway's host processor is communicating with LonWorks Neuron chip.</p>

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-NB 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 while 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 your 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.

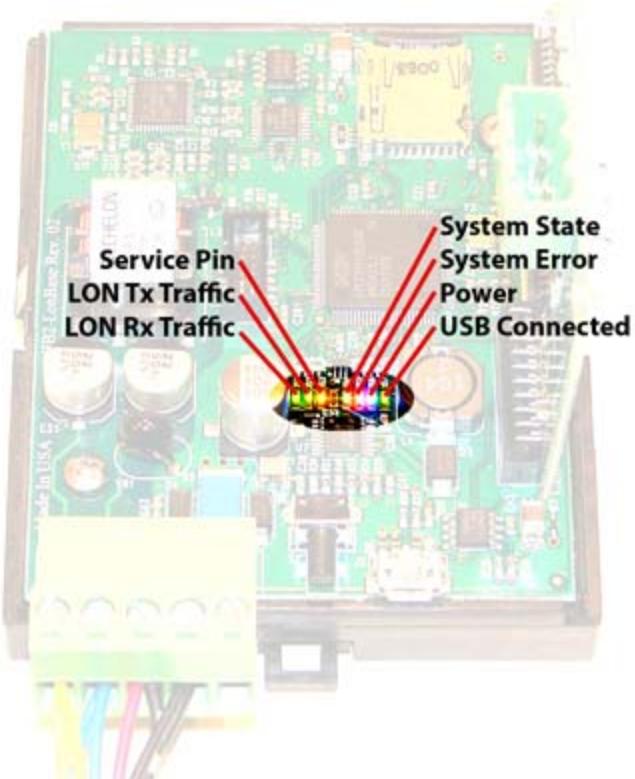
BB2-7020-NB IP LEDs		
Mode	IP Request LED	IP Reply LED
Gateway is Client (Master)	Flash yellow each time gateway sends a request to a remote server (slave).	Flash green when gateway receives a good response. Flash red when gateway receives error code message from server, or if timed out with no response 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.

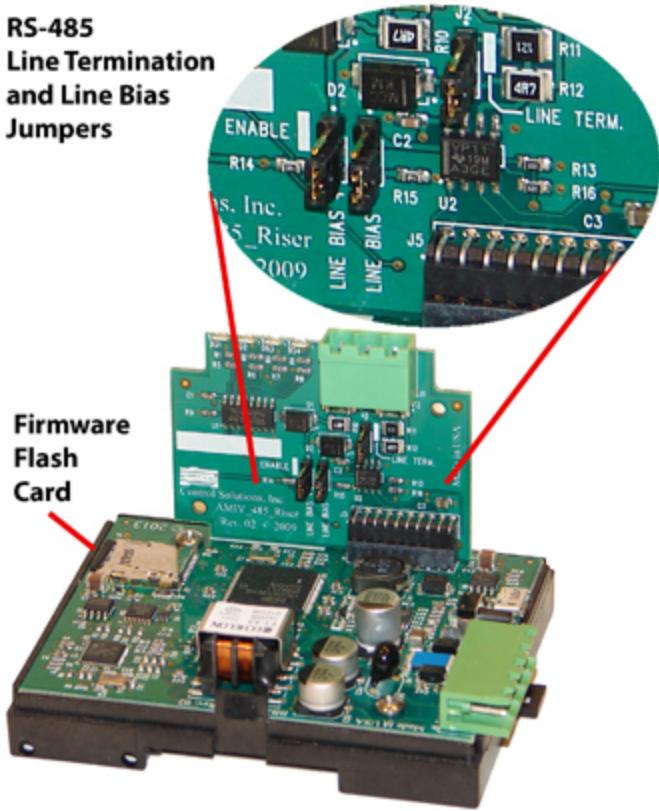
During power-up, the blue LED should turn on immediately, and most other LEDs will flash briefly. The system state and system error LEDs will deliberately flash once, and the system state LED will typically flash a short flash more than once before resuming normal heartbeat indication. If a hardware 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.

