

# ESSENTIALS



© Contemporary Control Systems, Inc.

## BACnet Discovery Tool Better than Ever but Still Free!

In an earlier *Essentials* article, we discussed the **BACnet Discovery Tool** (BDT) — a vendor-neutral tool to determine if a BACnet router is successfully communicating to attached devices. The present article updates that earlier one with several enhancements. The BDT functionality (version 2.03.00) has been expanded to easily and quickly report more detailed information. And it is still **FREE!**

BDT is a BACnet/IP application for Windows® that is easy to install and use. It is an excellent means for discovering and verifying communication with MS/TP devices that are being accessed through BACnet/IP routers such as those available from Contemporary Controls: the BASRT-B DIN-rail mounted BACnet/IP-to-MS/TP router — or its portable counterpart, the BASRTP-B. At Contemporary Controls we use BDT with equipment from different BACnet vendors to prove our routers operate with different equipment configurations.

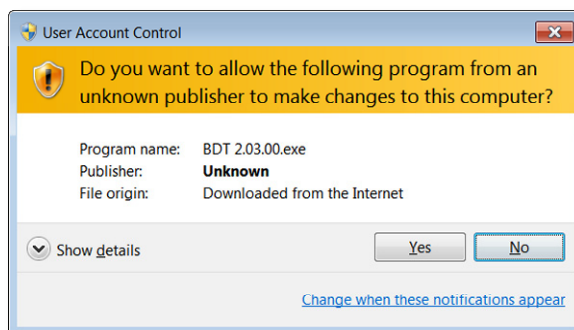
BDT has proved to be very popular — as one satisfied user reports:

*BDT 2.03 has become one of the BACnet applications I have begun to rely on when discovering BACnet devices and objects. For a quick snapshot of networks and the MAC addresses of the controllers on them is as easy as one click, and the support of BBMD to connect to remote sites is a great feature. The ability to scan the present values of all supported BACnet objects in discovered devices as well as the ability to write to the present value (at selectable priority levels) of certain objects make BDT 2.03 one of the most feature rich freely downloadable BACnet applications on the web.*

After downloading BDT, unzip its file set to any location on your host PC hard drive (be sure to keep all the files together at the chosen location). The file set will include an instruction sheet in PDF format and the following four files:

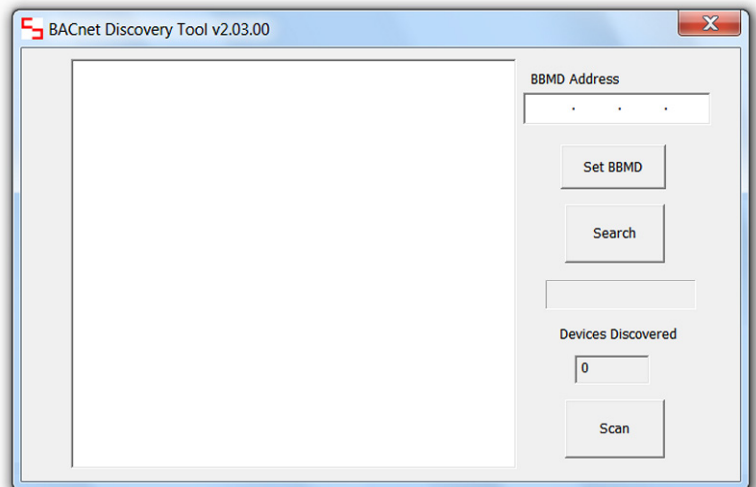
- bacnet-stack.dll
- BDT 2.03.00.exe
- mfc71d.dll
- msvcr71d.dll

If you are not using Windows 7, double-click on the **BDT2.03.00.exe** file to launch BDT. But if you are using Windows 7, right click the file name, choose to “Run as Administrator” and click Yes in the dialogue which appears to the right — otherwise, BDT will not function properly.



When you attempt to open BDT, Windows may notify you that the file has no valid digital signature. If so, it is safe to ignore this warning and proceed. As BDT starts, it opens in its own window similar to that shown below.

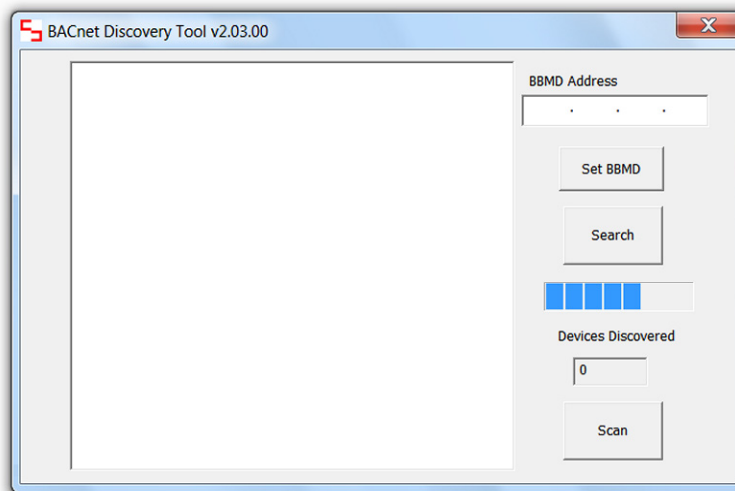
Before initiating any BDT activity, you should determine the scope of what you wish to accomplish. The first thing to consider is whether or not your BACnet internetwork crosses IP subnets — and is therefore using a BACnet/IP Broadcast Management Device (BBMD). If a BBMD is present, you may wish to use it to examine more of the BACnet internetwork than just your local subnet. You can do this by using the BDT's **Foreign Device Registration (FDR)** function to register the BDT application as a *Foreign Device* with a BBMD in a remote subnet.



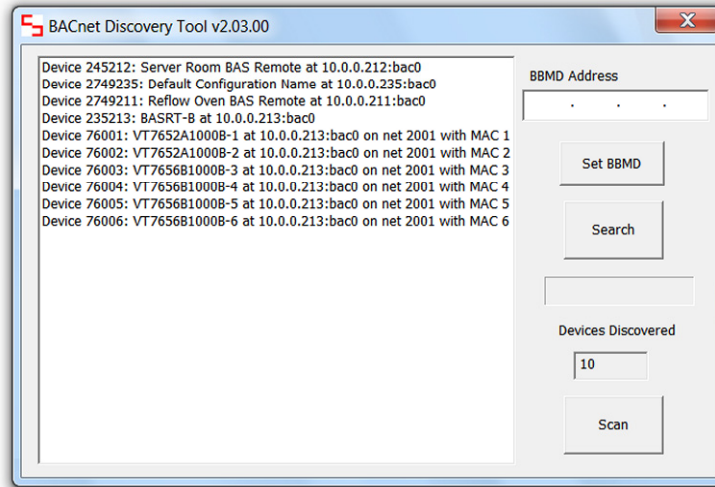
**Setting the BBMD, if needed.** In the *BBMD Address* field, enter the IP address of the BBMD with which you wish to register. (You do not specify a subnet mask — this is determined by other equipment in the network.) In some larger networks with several subnets, there may be multiple BBMDs. In such cases, you would normally register with the central BBMD and thus access the entire BACnet internetwork. You could perform FDR with a non-central BBMD on a specific subnet, but you may not know the extent to which the various BBMDs are sharing information — and you might not access the devices that you need to contact. Targeting the central BBMD is almost always the best option. When you click on the *Set BBMD* button, registration is completed.

**Using the Search function.** After setting the BBMD (if needed) and with your host PC attached to the network of interest, click the *Search* button — **not the Scan button**, which should only be used after a search has built a database of objects.

As the Search function runs, it transmits BACnet *Who-Is* messages and a progress bar appears as in the following screen. **NOTE:** BDT sends a **BACnet/IP Who-Is** — not a **BACnet/Ethernet Who-Is** — so it will not discover devices that only support BACnet/Ethernet.



When BDT has completed its search, the progress bar will disappear, the discovered BACnet devices will be listed in the main window and the number of discovered devices will be reported in the *Devices Discovered* field — similar to the display below.

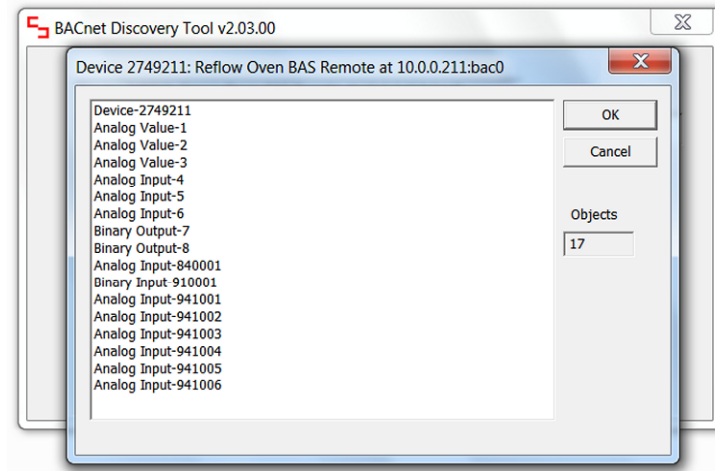


As shown in the sample screen above, each *I-Am* response identifies

- the *Device Instance* number of the responding device
- the *Device Name* of the device
- the *IP Address* and *UDP Port* number through which the device was contacted
- and (for MS/TP master devices) the MS/TP Network number and MS/TP *MAC Address*

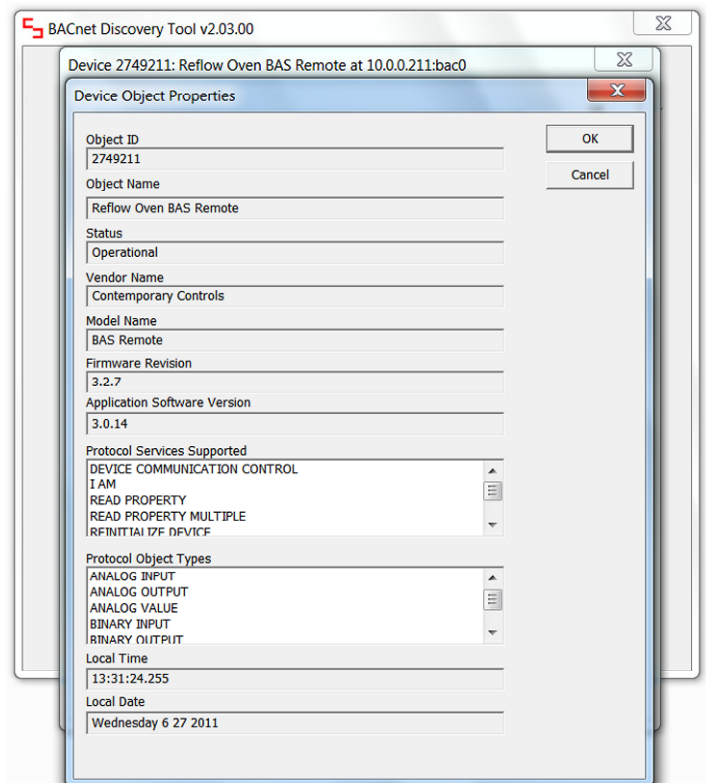
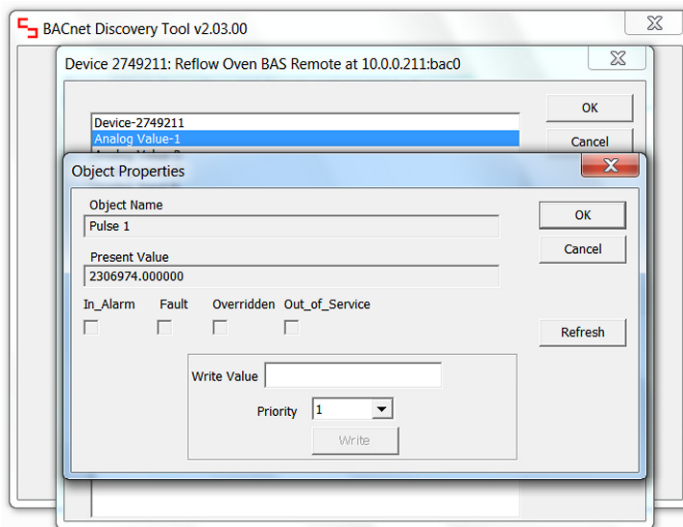
Each piece of equipment producing an *I-Am* response could be simply a BACnet/IP device (as is the case with the first three respondents in the screen above) — or it could be a BACnet/IP to BACnet MS/TP router acting as an intermediary for devices on the MS/TP side (which is the case with Device 235213 above). If the reported *Device Instance* is that of an MS/TP device, then the MS/TP Network and MAC information will complete the line.

To investigate the objects contained by any discovered device, double-click that device’s line in the discovered device list. As shown below, a new window will appear and display a list of the discovered **objects** within the selected device. Also, the number of discovered objects will be reported in the *Objects* field. BDT can support up to 1000 devices and each device can have up to 2000 objects.



You can double-click the device object line to examine the *Device Object Properties* (Device 2749211 in our example) which brings up a window similar to that which is displayed to the right. As shown, this window reports many helpful *Device Object Properties*.

If you close the window shown to the right and again display the list of objects contained by the device, you can double-click one of its objects to bring up an *Object Properties* window similar to that below.

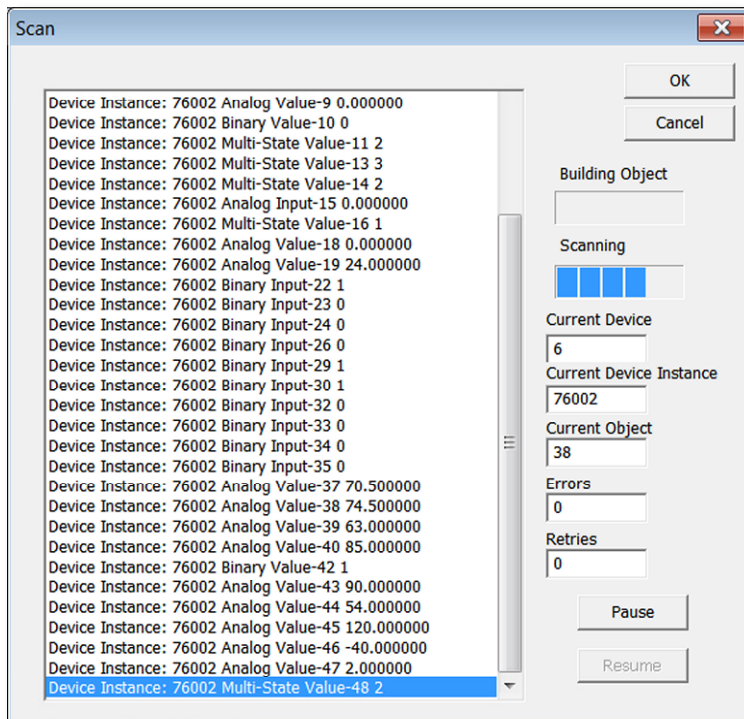
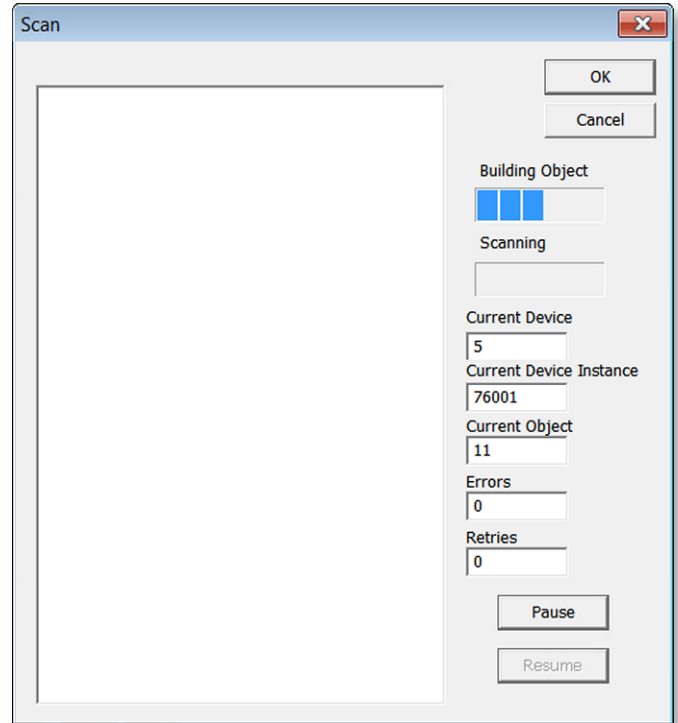


The *Object Properties* window displays read-only values — *Object Name*, its *Present Value* and four on/off status reports (checked means on) — and two writable elements lower down which, although always visible, only yield results for **output** objects. If the *Write* button is not dimmed, the examined object is writable — in which case, you can enter a number in the *Write Value* box, set a *Priority* value and click the *Write* button to apply your changes. Set a priority value by clicking the *Priority* field then adjusting the level displayed below the field by scrolling through the available levels using either your keyboard up/down arrow keys or the miniature up/down scroll controls. Clicking the *Refresh* button on the right will reacquire and re-display the read-only information.

After you are done with the *Object Properties* window, you can close it with either the *OK* or *Cancel* button. Then you can double-click another object to view and/or adjust its properties.

**Only click the Scan button after the Search function has been run.** Usually you should avoid the *Scan* function because it often creates more network activity than desired by *continuously* reading the *Present Value* of most objects that have this property (and alert you if an attempt fails). However, when commissioning or troubleshooting a network, the *Scan* function can usefully confirm that all devices and objects are properly installed, working and updating. This could be especially helpful if you do not have a building controller at your disposal to check this for you. **NOTE:** The *Scan* function only stops when you close the *Scan* window. Leaving it running can consume significant network bandwidth — and if a building controller is running, communication errors could result.

Executing the *Search* function creates an object database that is used by the *Scan* function. Clicking the *Scan* button opens a window similar to that below where you will see a bar indicating the progress in *Building Object* — that is, creating a scan list of objects extracted from the object database. As the list is built, the current *Device* and its *Object* are identified in their respective fields.



After the scan object list has been built, the scan proceeds as shown in the sample screen below. As the list is scanned, the current *Device* and its *Object* being scanned are identified in their respective fields. If an object fails to report its data, BDT retries the data acquisition four times before registering an error and moving on to the next object. These two fields — *Errors* and *Retries* — are incremented without reset for as long as the scan proceeds. You can also *Pause* the scan, if needed.

When scanning for present-values, BDT will display the currently read values. When all objects have been read, BDT will start displaying again — at the first line. Because present-values will appear in the same place in the display, current values can be viewed by scrolling the list up or down. If a present-value cannot be read, an error statement will occupy a line in the list.

When you are finished with scanning, click the *OK* button to terminate the scan. When you are finished with the BDT, close all of its windows. You can download this handy, free application by pointing your browser to:

[www.ccontrols.com/sd/bdt.htm](http://www.ccontrols.com/sd/bdt.htm)