

**NISTIR 7825**

# **Using the BACnet Data Source v1.7**

Michael A. Galler

**NISTIR 7825**

# **Using the BACnet Data Source v1.7**

**Michael A. Galler**

*Mechanical Systems and Control Group  
Building Environment Division  
Engineering Laboratory  
National Institute of Standards and Technology  
Gaithersburg, MD 20899-8631*

September 2011



**U.S. DEPARTMENT OF COMMERCE**

*Rebecca Blank, Acting Secretary*

**NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY**

*Patrick Gallagher, Director*

## **Preface**

The BACnet Data Source (BDS) was developed to be used as a component of the Virtual Cybernetic Building Testbed (VCBT) [1], in conjunction with research on various Fault Detection and Diagnostics (FDD) [2] and Building Commissioning (Cx) tools. The VCBT uses real BACnet enabled mechanical systems controllers coupled with computer simulations to emulate an entire building, mixing the real controllers with computer simulations representing the rest of the building. The BDS communicates with the controllers in the VCBT to retrieve data and can make the data available to FDD or Cx tools in real-time through an Open Database Connectivity (ODBC) connection, or can save the data to a file for later analysis. By immediately sending the data to the FDD or Cx tools in real-time, the BDS allows the tools to analyze data from a building in real-time and alert the operator of any problems immediately. The BDS retrieves data from the controllers by communicating with them through BACnet, using the BACnet Communications DLL (BCD) [3]. This paper will discuss how to use the BDS with the data saved to files. Configuring an ODBC connection is beyond the scope of this paper.

## **Disclaimers**

Any mention of commercial products in the BDS or this user's guide is for information purposes only; it does not imply recommendation or endorsement by NIST.

This software was developed at the National Institute of Standards and Technology by employees of the Federal Government in the course of their official duties. Pursuant to Title 17 Section 105 of the United States Code this software is not subject to copyright protection and is in the public domain. This software is an experimental system. NIST assumes no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic.

This software can be redistributed and/or modified freely provided that any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified. We would appreciate acknowledgement if the software is used.

## Table of Contents

1. Introduction to the BACnet Data Source .....	1
2. Introduction to the BDS User Interface .....	2
2.1. Section 1 of the BDS Interface- Command Buttons.....	2
2.2. Section 2 of the BDS Interface- Device and Datapoint Displays.....	4
2.3. Section 3 of the BDS Interface- Real-time Data Display.....	7
2.4. Section 4 of the BDS Interface- Configuration File Selection .....	7
2.5. Section 5 of the BDS interface- Log Files and Databases.....	8
2.6. Section 6 of the BDS Interface- Status Display .....	9
2.7. Section 7 of the BDS Interface- Network and Time Information .....	9
2.8. BDS Menu Functionality.....	10
3. Using the BDS- Use Cases .....	12
3.1. Use Case 1: Starting the BDS the first time- network discovery. ....	12
3.2. Use Case 2: Adding elements to the datapoint list. ....	12
3.3. Use Case 3: Loading a configuration file. ....	13
3.4. Use Case 4: Changing the RPM value for a device.....	14
3.5. Use Case 5: Saving datapoints to multiple files. ....	14
4. Configuration Files .....	16
5. Details of data file operations .....	19
6. References.....	20

## Table of Figures

Figure 1- Diagram of BDS functionality. ....	1
Figure 2- Sections of interest on the BDS. ....	2
Figure 3- Section 1 at startup. ....	3
Figure 4- Section 1 after BDS initialization. ....	3
Figure 5- Section 2 in Device mode. ....	5
Figure 6- Section 2 in Datapoint mode. ....	5
Figure 7- Section 3 with datapoints found on network.....	7
Figure 8- Section 3 with datapoint labels retrieved or read in from a settings file.....	7
Figure 9- Section 4 with sample file names.....	8
Figure 10- Section 5 with sample file names and directories. ....	8
Figure 11- Timer boxes under Section 5.....	9
Figure 12- Section 6 at startup. ....	9
Figure 13- Section 7 at startup. ....	10
Figure 14- The File menu. ....	10
Figure 15- The Settings menu.....	10
Figure 16- Changes to the display when adding a new output file. ....	15
Figure 17- Sample configuration file.....	16
Figure 18- Datapoints configured in BDS. ....	19
Figure 19- The same datapoints saved to a CSV formatted file. ....	19

## 1. Introduction to the BACnet Data Source

The BACnet Data Source (BDS) fundamentally acts as a datalogger. The BDS retrieves data from BACnet enabled controllers by sending BACnet requests to the controllers. The requests can be for any property of any object on the controller, but are commonly restricted to the Present\_Value property of Analog Input (AI), Analog Output (AO), Analog Value (AV), Binary Input (BI), Binary Output (BO), and Binary Value (BV) objects, or for the Local\_Date or Local\_Time properties of a Device object.

The list of monitored points (datapoint list) retrieved by the BDS may be set manually through the interface, or it may be read from a configuration file. The datapoint list is central to the functionality of the BDS. A useful feature of the BDS is that subsets of the datapoints can be saved into separate output files, and any single datapoint can be saved into more than one output file. The data set sent to the optional Open Database Connectivity (ODBC) connection can also be a subset of the datapoints collected. The BDS can monitor several thousand datapoints simultaneously. The BDS can retrieve data from multiple controllers at the same time, as well as communicating via Ethernet and BACnet/IP to different controllers.

The BDS is capable of recording values from BACnet objects on BACnet speaking mechanical systems controllers and passing them to a database through an ODBC connection, and also saving time series data into files for later analysis. This functionality is diagrammed in Figure 1.

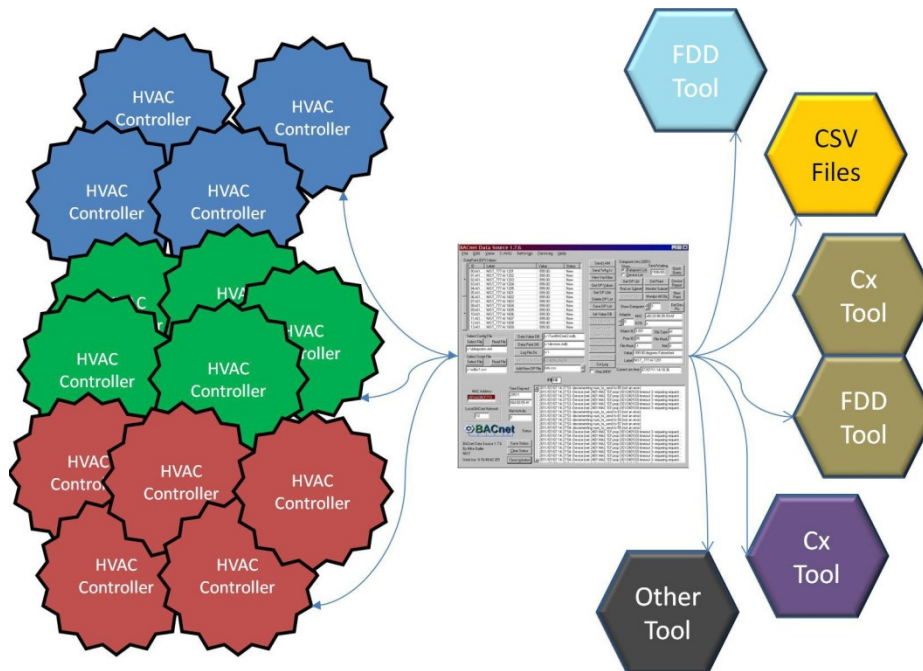


Figure 1- Diagram of BDS functionality.

The BDS uses WinPcap, which is available from <http://www.winpcap.org>, and the BACnet Communications DLL (BCD), which is available from the author at NIST. WinPcap should be installed first. The BDS and the BCD files may then be copied to any directory and run from there.

## 2. Introduction to the BDS User Interface

Interaction with the BDS occurs through the graphical user interface (GUI). The GUI contains several main sections, each with a separate purpose. The sections are shown in Figure 2.

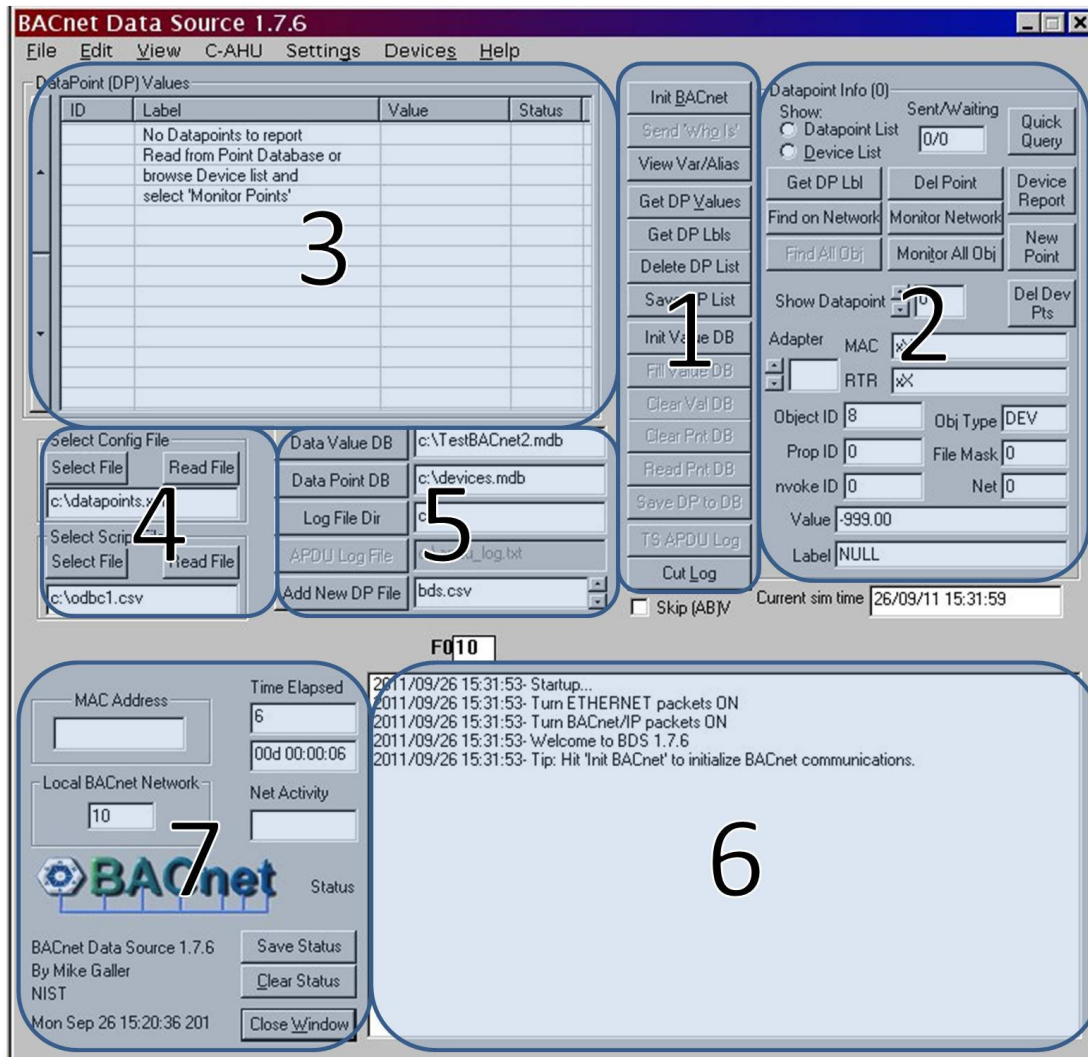


Figure 2- Sections of interest on the BDS.

### 2.1. Section 1 of the BDS Interface- Command Buttons

Section 1 contains buttons that allow the user to initialize the BDS and interact with the network, the list of monitored points (datapoint list), the database, and the log file containing data. The function and availability of some of the buttons will change depending on the mode the BDS is in, as shown in Figure 3 and Figure 4. A description of the function and use of the buttons follows:

**Init BACnet-** Initializes BACnet communications subsystem. After initialization, the button text changes to ‘Send I-AM’.

**Send I-AM-** Broadcasts a BACnet I-Am message on the network.

**Send ‘Who Is’-** Broadcasts a BACnet Who-Is message on the network. Responses from other devices on the network can be reviewed by cycling through the device list.

**View Var/Alias-** Displays a list of user defined variables and aliases in the status window. Variables and aliases are discussed in Section 4 of this paper, and the status window is discussed in Section 2.6.

**Get DP Values-** Retrieve the values listed in the datapoint list. The BDS will send BACnet ReadPropertyMultiple service requests to the appropriate devices.

**Get DP Lbls-** Retrieves the value of the Object\_Name property for every entry in the datapoint list, and saves it as the label. This will overwrite the current labels.

**Delete DP List-** Empties the datapoint list.

**Save DP List-** Saves the datapoint list to a settings file. The settings file may be used to configure the datapoint list on subsequent uses of the BDS. The file, and how to use it, are discussed in more detail in Section 2.4 of this paper.

**Init Value DB-** Initialize the database. Use of the database is optional. Configuring the database in the BDS is outside the scope of this paper.

**Fill Value DB-** Copies values from the datapoint list into the database.

**Clear Val DB-** Clears all values written to the database.

**Clear Pnt DB-** Clears all datapoint information from the database.

**Read Pnt DB-** Displays datapoint information from the database in the status window.

**Save DP to DB-** Saves datapoint list information to the database.

**TS APDU Log-** Timestamp the application protocol data unit (APDU) log file, by appending the date and time to the file name. Subsequent data are written to a new file with the default name. The APDU log file is discussed in more detail in Section 2.8.

**Cut Log-** Timestamp data files, by appending the date and time to the file name. Subsequent data are written to a new output file with the default name. New column header information is also written to the new file. The output files are discussed in more detail in Section 5.

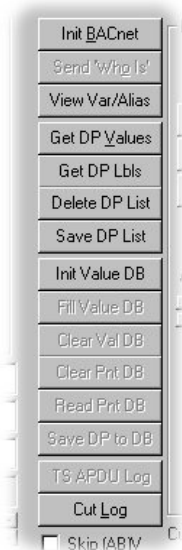


Figure 3- Section 1 at startup.



Figure 4- Section 1 after BDS initialization.

## 2.2. Section 2 of the BDS Interface- Device and Datapoint Displays

Section 2 contains buttons and text boxes that allow the user to interact with BACnet devices on the network, and to view and modify the list of datapoints and the list of devices on the network. The controls in Section 2 operate in one of two modes, Datapoint List mode or Device List mode. Selecting the radio button labeled “Datapoint List” or “Device List”, respectively, sets the mode. As in Section 1, the function of some of the display elements will change depending on the mode the section is in. The two modes are shown in Figure 5 and Figure 6. In both modes, the user is able to scroll through the list of devices or datapoints, with the information displayed in the text boxes in the bottom half of the section. The buttons in the top section allow the user to perform an action related to either the specific item being viewed, or to the list as a whole. The function and use of the display elements in Section 2 is described below for Device List mode:

**Found Device Info-** Label for the box surrounding this section. Displays count of items in the device list.

**Show Device-** Clicking on the up or down arrows next to this label allows the user to cycle through the list of devices. The text box next to the up and down arrows displays a number representing the placement of the device in the list. The information in the text boxes below will change as the currently selected device is changed.

**MAC-** Displays the Media Access Control (MAC) address of the currently selected device.

**RTR-** Displays the Router IP or MAC address, if a router is present.

**Object ID-** Displays the BACnet Device Object\_Identifier of the currently selected device.

**Prop ID-** Not used in this mode.

**RPM cnt-** Displays the count of the number of Property\_Identifiers requested in a ReadPropertyMultiple service request to this device. This determines the size of the packets sent when requesting data. This number can be modified by typing a new number in the box and then clicking outside of the box. If this value is too large, and the controller does not support segmentation, then a controller will not be able to send a response to the request, and will instead generate an error message.

**Objects-** Displays a summary of information about the BACnet objects present on the currently selected device, including the total count and the number of each type present.

**Label-** Displays a text label associated with the device. The BDS will include manufacturer information in the label if possible.

**Net-** Displays the BACnet network number for the currently selected device.

**Find Objects-** When pressed, the BDS will send BACnet messages to retrieve the list of objects on the currently selected device. The BDS will use ReadPropertyMultiple service requests to retrieve the information. Some devices may need to have the ‘RPM cnt’ parameter adjusted to perform well if



the default value results in packets that are too large. The BDS will reduce this parameter automatically if necessary.

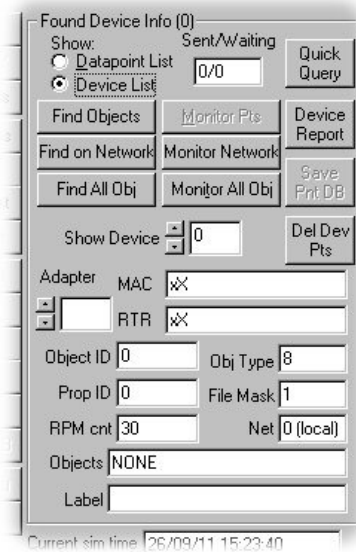


Figure 5- Section 2 in Device mode.

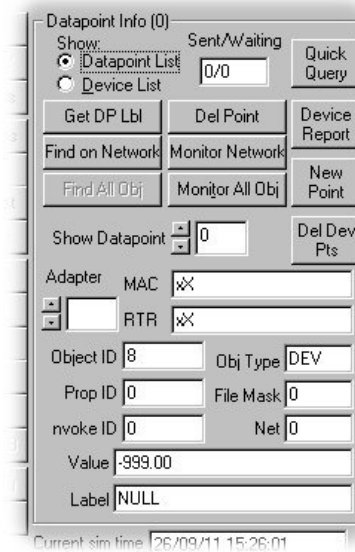


Figure 6- Section 2 in Datapoint mode.

**Monitor Points-** Appends all AI, AO, AV, BI, BO, and BV objects on the currently selected device to the datapoint list. AV and BV objects may be excluded by selecting the checkbox below Section 1 labeled ‘Skip (AB)V’.

**Find on Network-** Performs the ‘Find Objects’ function for every known device on the same BACnet network as the currently selected device.

**Monitor Network-** Performs the ‘Monitor Points’ function for every known device on the same BACnet network as the currently selected device. If a device has not been queried and has no known objects, it will be skipped.

**Find All Obj-** Performs the ‘Find Objects’ function for every device in the device list.

**Monitor All Obj-** Performs the ‘Monitor Points’ function for every device in the device list. If a device has not been queried and has no known objects, it will be skipped.

**Quick Query-** Retrieves Object\_Name property and Object\_List count from every device in the device list. These data populate the Label and Objects fields in the BDS display.

**Device Report-** Saves the device list to a file in an Extensible Markup Language (XML) format. This also saves any configuration information that has been retrieved from the controllers on the device list. This file may be used in future BDS sessions to populate the device list. The procedure to read configuration files is discussed in Section 4.

**New Point-** Allows the user to create a new datapoint. If there is a currently selected datapoint, the new one will be a duplicate. If there are no datapoints, the configuration information will have to be entered manually.

**Del Dev Pts-** Deletes items from the datapoint list which are located on the currently selected device.

The function and use of the display elements in Section 2 in Datapoint List mode is described below:

**Datapoint Info-** Label for group box surrounding section. Displays count of items in the datapoint list.

**Show Datapoint-** Clicking on the up or down arrows next to this label allows the user to cycle through the list of devices. The text box next to the up and down arrows displays a number representing the placement of the device in the list.

**MAC-** Displays the MAC address of the device the currently selected datapoint is located on. This is a unique identifier associated with the network card on each device.

**RTR-** Displays the Router IP or MAC address, if a router is present.

**Object ID-** Displays the BACnet Device Object\_Identifier of the device on which the currently selected datapoint is located.

**Prop ID-** Displays the property identifier of the BACnet property retrieved. The value should not be changed from 85, Present\_Value. While the BDS can read other properties, it will not process them properly as part of the datapoint list.

**Invoke ID-** Displays the BACnet invoke-ID last used to retrieve the value associated with this datapoint. This is used for debugging problems with the BDS.

**Obj Type-** Displays the BACnet object type of the currently selected datapoint. This will be one of AI, AO, AV, BI, BO, or BV.

**File Mask-** Used to determine which output files the currently selected datapoint should be saved to. Use of the file mask is discussed in more detail in Section 3.5.

**Net-** Displays the BACnet network number for the device containing the currently selected datapoint.

**Value-** Displays the last value read for the currently selected datapoint. Units will also be displayed if they are available from the device. If the value has not been read, the value -999 will be shown. To write a new value to the datapoint, erase the text in the display, enter the new number, and click outside of the Value box.

**Label-** Displays a text label associated with the datapoint. The BDS will generate a generic label based on the device label and the type of datapoint. This is done when the datapoint is generated.

**Get DP Label-** Retrieves the Object\_Name property from the currently selected datapoint. This will overwrite the current Label.

**Find on Network-** Performs the 'Find Objects' function for every known device on the same BACnet network as the currently selected datapoint.

**Monitor Network-** Performs the 'Monitor Points' function for every known device on the same BACnet network as the currently selected datapoint. If a device has not been queried and has no known objects, it will be skipped.

**Monitor All Obj-** Performs the ‘Monitor Points’ function for every device in the device list. If a device has not been queried and has no known objects, it will be skipped.

**2.3. Section 3 of the BDS Interface- Real-time Data Display**

Section 3 contains a real-time display of data read from BACnet controllers. The display is separated into four columns, as shown in Figure 7. Each row contains information about one member of the datapoint list. The selection displayed can be changed by clicking on the up or down arrows to the left of the display, or by typing a number in the ‘Show Datapoint’ box in Section 2 and typing in the ID number of the datapoint to be displayed.

The ID column contains two types of information, the datapoint ID number, and BACnet object type and instance information. The datapoint ID number represents the location of a datapoint in the array containing all datapoints. This number can be typed in the ‘Show Datapoint’ text box in Section 2 to change the display to show a particular datapoint, which is useful when there is a large number of datapoints. The BACnet object type and instance information shows the type of object (i.e., Analog Input, Binary Output ...) and the instance number of that object on the host device. A colon separates these two fields in the display. The first datapoint in the list has ID 0. If the first datapoint in the list is an Analog Input (AI) object with an instance number of 1, the ID field for that datapoint would read “00:AI01”.

The Label column contains descriptive text. When datapoints are added to the datapoint list, a default label will be generated consisting of the label of the host device, the type of BACnet object that a datapoint references, and the instance of that object on the host device. This data can be retrieved from the host device by clicking on the ‘GetDP Lbls’ button in Section 2. An example of labels read from a configuration file or retrieved from the host device is shown in Figure 8.

ID	Label	Value	Status
00:AV00	Device 100 AV 0	-999.00	New
01:AV01	Device 100 AV 1	-999.00	New
02:AV02	Device 100 AV 2	-999.00	New
03:AV03	Device 100 AV 3	-999.00	New
04:AV04	Device 100 AV 4	-999.00	New
05:AV05	Device 100 AV 5	-999.00	New
06:AV06	Device 100 AV 6	-999.00	New
07:AV07	Device 100 AV 7	-999.00	New
08:AV08	Device 100 AV 8	-999.00	New
09:AV09	Device 100 AV 9	-999.00	New
10:AV10	Device 100 AV 10	-999.00	New
11:AV11	Device 100 AV 11	-999.00	New
12:AV12	Device 100 AV 12	-999.00	New
13:AV13	Device 100 AV 13	-999.00	New

Figure 7- Section 3 with datapoints found on network.

ID	Label	Value	Status
00:AI00	SA Temp	-999.90	New
01:AI01	SF Pressure	-999.90	New
02:AI02	RA Temp	-999.90	New
03:AI03	RA Humid	-999.90	New
04:AI04	OA Temp	-999.90	New
05:AI05	OA Humid	-999.90	New
06:AI06	SA Flow CFM	-999.90	New
07:AI07	RA Flow CFM	-999.90	New
08:AI08	MA Temp	-999.90	New
09:A000	SF Speed	-999.90	New
10:A001	CC Valve	-999.90	New
11:A002	HC Valve	-999.90	New
12:A003	MAMP	-999.90	New
13:A004	RF Speed	-999.90	New

Figure 8- Section 3 with datapoint labels retrieved or read in from a settings file.

**2.4. Section 4 of the BDS Interface- Configuration File Selection**

Section 4 allows the BDS to read files containing saved datapoint lists, information about the BDS configuration, data about the local network, and scripts. Details about the format of these files are described in Section 4. Files containing datapoint lists or information about the BDS configuration or the local network can be entered either by typing the name into the textbox in the ‘Select Config File’ area as shown in Figure 9, or by clicking on the ‘Select File’ button above the textbox, and then clicking on the ‘Read File’ button. Script files can be entered by

using the textbox in the ‘Select Script File’ area or the ‘Select File’ button, and then clicking the ‘Read File’ button. Multiple settings and script files may be entered during one BDS session.

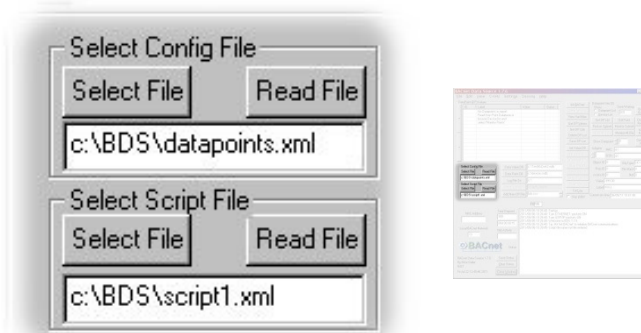


Figure 9- Section 4 with sample file names.

### 2.5. Section 5 of the BDS interface- Log Files and Databases

In Section 5 the user may enter file names for databases, the log file, may edit the names of the output data files, and add new output data files. The various components of Section 5 are shown in Figure 10.



Figure 10- Section 5 with sample file names and directories.

The values of the datapoints stored in the BDS can be saved to a database, as described in Section 1. The database used to store values can be specified by clicking the ‘Data Value DB’ button, and selecting a file or creating a new one in the resulting dialog box. Similarly, the database used to store data point information can be specified by clicking on the ‘Data Point DB’ button.

The output data files, containing data from the datapoint list (usually time-series data the BDS has read from the controllers) are saved in the directory specified in the text box next to the ‘Log File Dir’ button. Clicking on that button allows the user to select a new folder for the data files. To add a new data file, click on the ‘Add New DP File’ button. The contents of text box next to the button will change to the default name for a new file, “DataFile0X.csv” where X is the number of the file, i.e., 1, 2, 3, etc. A new name may be entered if desired. Additionally, a new file timer box will appear below Section 5. The timer box displays the interval, in seconds, at which the data file is saved. It also allows the user to modify the interval by typing a new number into the appropriate box. The timer boxes are shown in Figure 11.

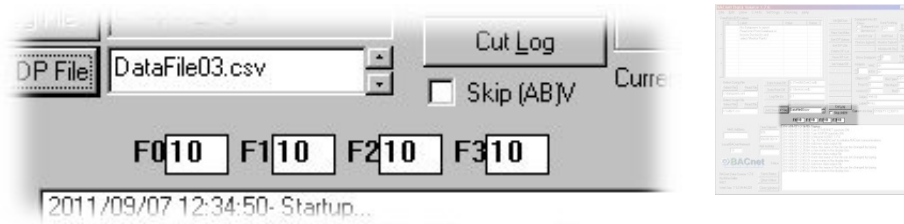


Figure 11- Timer boxes under Section 5.

## 2.6. Section 6 of the BDS Interface- Status Display

Section 6 is a display of diagnostic information, status updates, and other information from the BDS. Each line begins with a date and time stamp, followed by the content of the message. The contents of this status window may be saved to a text file by clicking on the 'Save Status' button to the left of the status window. The status window can be cleared by clicking on the 'Clear Status' button, also to the left of the status window. The status window is shown in Figure 12.

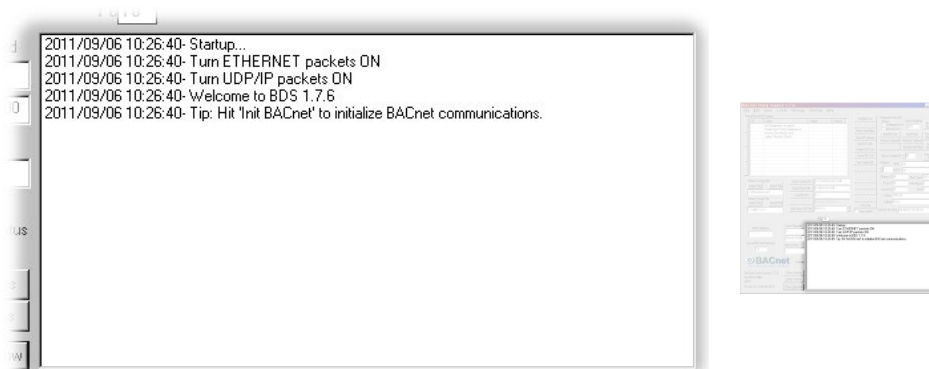


Figure 12- Section 6 at startup.

## 2.7. Section 7 of the BDS Interface- Network and Time Information

Section 7 contains text boxes displaying information relating to network operations and time. Information displayed includes the MAC address of the Network Interface Card (NIC) currently selected, as described in Section 2; the local BACnet network; a count of packets recently received; the time elapsed since the BDS was started, in seconds; and the time elapsed in 'DD hh:mm:ss' format. It also contains two buttons relating to the status window, as described in Section 3.6. There is also a button labeled 'Close Window'. Pressing this button will cause all output files to be flushed, all modified values on controllers to be reset, and the BDS to close. Section 7 is shown in Figure 13.

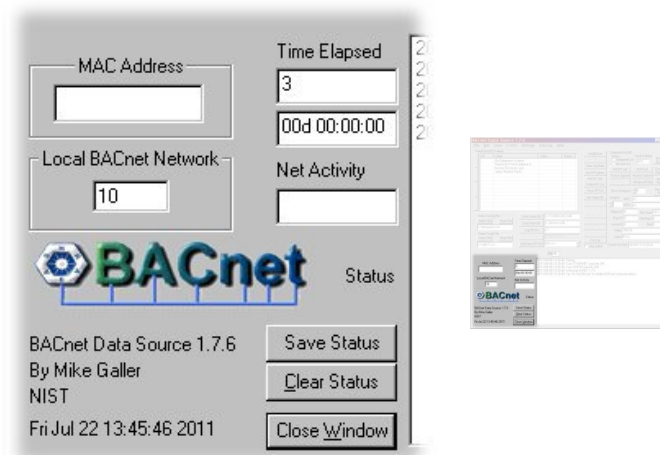


Figure 13- Section 7 at startup.

## 2.8. BDS Menu Functionality

The BDS also has many functions available through the menus located at the top of the BDS window. This section will explain this functionality.

The File menu, shown in Figure 14, allows the user to open a configuration file or a script file. These files can also be opened as described in the discussion of Section 4. The 'Save Enumerations' option allows the user to save a list of BACnet enumerations known by the BDS. This can provide a useful check when working with new BACnet functionality. The other items on the menu are awaiting functionality.

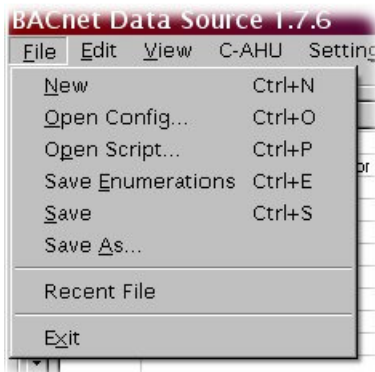


Figure 14- The File menu.

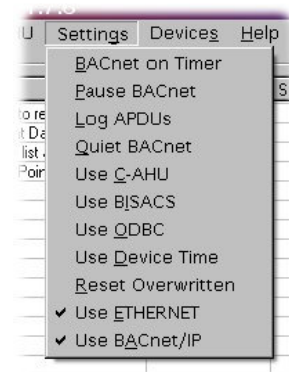


Figure 15- The Settings menu.

The View menu contains the 'Spin Data' option. Enabling this will cause the data display in Section 3 to continuously scroll through the entire set of values.

The Settings menu contains commands relating to data collection, retention, and status. The options are shown in Figure 15. The commands are:

**BACnet on Timer-** Selecting this starts automatic data collection. The items in the datapoint list will be retrieved on an interval determined by the timer for the first data file. Data will be written to output files at the interval specified for each file.

**Pause BACnet-** This may be selected to temporarily pause BACnet requests.

**Log APDUs-** This allows Application Layer Protocol Data Units (APDUs) received from the network to be logged to a file. The file may then be opened and examined using a BACnet protocol analyzer.

**Quiet BACnet-** This reduces the amount of information shown in the status window.

**Use C-AHU-** Obsolete, retained for backwards compatibility. This will be removed in a future version of the BDS.

**Use BISACS-** Obsolete, retained for backwards compatibility. This will be removed in a future version of the BDS.

**Use ODBC-** Configuration of the ODBC interface is beyond the scope of this paper.

**Use Device Time-** Reads the time from the device associated with the first item in the datapoint list, and uses that for display and timestamps.

**Reset Overwritten-** The BDS allows the user to modify the Present\_Value property of commandable BACnet objects located on controllers accessible to the BDS. This is done by writing to a Present\_Value property with a Priority\_Array value of 8 (Manual Operator). Selecting this command will relinquish all priority 8 slots modified by the BDS. Values are also relinquished automatically when the BDS exits.

**Use ETHERNET-** Enables BACnet communications using the Ethernet protocol. This is enabled by default. When deselected, the BDS will ignore all Ethernet packets received.

**Use BACnet/IP-** Enables BACnet communications using the BACnet/IP protocol. This is enabled by default. When deselected, the BDS will ignore all BACnet/IP packets received.

The Devices menu allows the user to scroll through the device list, using the display in Section 2. There are two options, 'Display Next' and 'Display Prev', which will change the display to the next or previous item in the device list. These have the same effect as selecting the up or down arrows next to the 'Show Device' label in Section 2. If Section 2 is displaying information from the Datapoint list, this menu will scroll through the Datapoint list.

### 3. Using the BDS- Use Cases

This section instructs the user in how to use the BDS through the examples given by Use Cases, as suggested by ASHRAE Guideline 20-2010 [4]. Various interactions a user could initially or commonly have with the BDS are described in the use cases presented here. This is not an exhaustive list, but a set chosen to be sufficient to allow a new user to become familiar with and use the BDS. The use cases described here are:

- Starting the BDS the first time- network discovery.
- Adding elements to the datapoint list.
- Loading a configuration file.
- Changing the RPM value for a device.
- Saving datapoints to multiple files.

#### 3.1. Use Case 1: Starting the BDS the first time- network discovery.

When using the BDS for the first time, the user will not have a configuration file or information about the local configuration. The easiest method to obtain this information is to have the BDS query the network to discover the local devices and configuration. After the BDS is started, complete the following activities:

1. Click on the 'Init BACnet' button in Section 1. This will initialize communications with the BACnet Communications DLL (BCD). Any messages from the BCD will be displayed in the status window. If your computer has more than one network card installed, check that the BDS is using the correct card. Note that the BDS will try to skip over virtual network cards. If necessary, change the default adapter as detailed in the overview of Section 2.
2. Optionally adjust the 'Use Ethernet' and 'Use BACnet/IP' options in the Settings menu.
3. Click on the 'Send Who Is' button in Section 1. This will send a BACnet Who-Is broadcast message on the local network. The responses to this request will automatically be added to the device list. Note that the BDS will parse any BACnet I-Am packets received and add new devices to the list, even if the packet was not generated as a response to a request from the BDS.
4. Wait for the local devices to respond. This should only take a few seconds, except for very large networks. You may examine the list at any time, as detailed in the overview of Section 2.
5. When all devices have responded, this use case is completed. Subsequent steps include saving the configuration to a file by clicking on the 'Device Report' button in Section 2, or an action described by another use case.

#### 3.2. Use Case 2: Adding elements to the datapoint list.

Without adding elements to the datapoint list, the BDS has minimal usefulness. This use case has as a prerequisite that the BDS has a device list, either through completion of Use Case 1, or through loading of a configuration file. Note that the configuration file may have been created by the BDS through using the 'Device Report' button or the 'Save DP List' button, or may have



been generated manually or by another tool. To add elements to the datapoint list, complete the following activities:

1. Scroll through the device list, either by clicking the arrows next to 'Show Device' in Section 2 or by entering a number in the text box next to the arrows. See the overview of Section 2 for more details.
2. When a device of interest is displayed, retrieve information from it by clicking on the 'Find Objects' button, as detailed in the overview of Section 2. If there are other devices of interest on the same BACnet network, click on the 'Find on Network' button to query all devices on that BACnet network. If all devices reachable on the network are of interest, click on the 'Find All Obj' button to retrieve information from all devices that have been found.
3. When all responses are received, click on the 'Monitor Pts' button to add all of the datapoints from the currently displayed device. Alternatively, click on the 'Monitor Network' or 'Monitor All Obj' buttons if appropriate. The BDS will add all discovered elements of interest to the datapoint list. Elements of interest are Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, and Binary Value objects on the device(s) of interest. To exclude Analog Value and Binary Values from the list, select the 'Skip (AB)V' checkbox located below Section 1.
4. More datapoints may be added to the list by scrolling the display to a different device and repeating the above steps. If using the 'Monitor Network' button, make sure that subsequent devices are on different BACnet networks to avoid duplication in the datapoint list.
5. When the datapoints from all devices of interest have been added to the datapoint list, this use case is completed. Subsequent steps include saving the configuration by clicking on the 'Save DP List' button, or initiating data collection by selecting 'BACnet on Timer' in the Settings menu.

### **3.3. Use Case 3: Loading a configuration file.**

This method will probably be most commonly used to initiate data collection using the BDS. This use case has as a prerequisite that a configuration file exists, either from a previous use of the BDS or from creating one manually. To load a configuration file, complete the following activities:

1. After starting the BDS, click on the 'Select File' button in the 'Select Config File' area. This can be found in Section 4 of the BDS display. Clicking on this button will open a standard interface allowing the user to select a file.
2. Select the settings file of interest. The directory structure may have to be traversed to find the desired file. Click on the 'Open' button in the interface.
3. Alternatively, if the name/location of the file is known, it can be typed into the text box below the 'Select File' button.
4. Once the file name/location has been entered, click the 'Read File' button to have the BDS open and process the file. The BDS will read in the information in the file, including devices, datapoints, output file names, and the log file directory. After the file is read, the BDS will enable the 'BACnet on Timer' option, found in the Settings menu.

This will initiate data collection by enabling the BDS to retrieve data as indicated in the datapoint list, and save it to the output files.

5. Check the status window for any errors or other messages. If the configuration file has been loaded successfully, this use case is completed.

### **3.4. Use Case 4: Changing the RPM value for a device.**

Some devices may not be able to communicate at the default setting the BDS uses for ReadPropertyMultiple (RPM) BACnet messages. This may manifest as a device or multiple devices on a BACnet network appearing partially or non-responsive. The solution to this problem is to reduce the RPM count, which sets the number of ReadAccessSpecifications sent in each packet. Each ReadAccessSpecification corresponds to one element of the datapoint list. The RPM value is changed by completing the following activities:

1. After starting the BDS, load the device list using one of the methods previously outlined.
2. Scroll the device display to the device of interest.
3. Locate the text box next to the 'RPM cnt' label. This is in Section 2.
4. Change the value in the text box. Type a new number in the box and hit the tab key to move focus to a different text box. If the number entered is out of range (3 - 40), the previous value will be restored. The default value is 30, but it will be different if it has been changed previously. Lower values are appropriate for a device with a small maximum APDU size. Some configurations may require values as low as 3.
5. Test the device of interest. If it is not responding at an acceptable level, repeat the previous step with a lower value. If the device is responding at an acceptable level, then this use case is completed. These steps may need to be repeated for other similar devices. After all devices have been reviewed, it is recommended to save the configuration file so that the changes will remain in effect for subsequent runs.

### **3.5. Use Case 5: Saving datapoints to multiple files.**

The ability to route the output data of the BDS to multiple targets is a very useful feature. Each datapoint can be sent to one or more destinations, usually a CSV file. This use case has as a prerequisite that a configuration file with datapoints exists. The activities to perform to modify the output files that one or more datapoints are sent to are:

1. After starting the BDS, load the datapoint list using one of the methods previously outlined.
2. Add a new output file. Click on the button labeled 'Add New DP File' in Section 5 of the BDS interface. For the purposes of this example, it is assumed that only one datapoint file was defined previously. The textbox next to the button should change to read 'DataFile01.csv', and a new box should appear below with the label 'F1' and a value '10' in it. These changes are shown in Figure 16. You may optionally type a new name for the new output file.
3. Put the BDS Section 2 display in datapoint mode by selecting the 'Datapoint List' radio button.

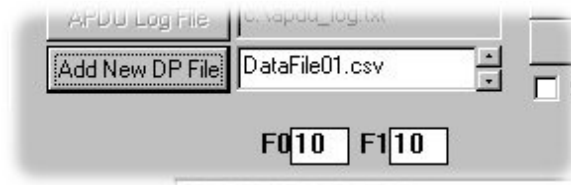


Figure 16- Changes to the display when adding a new output file.

4. Navigate the display to show a datapoint that is to be added to the new file. This is done by clicking the up and down arrows next to the 'Show Datapoint' label in Section 2 of the display. The number displayed in the text box next to the arrows will change with the selection. If the number of the datapoint is known, it may be typed directly into the textbox. This is useful if there is a large number of datapoints. After typing the desired number, the cursor must be moved out of the text box, either by hitting the tab key or selecting another area with the mouse. The BDS will not read any text box entries until this is done.
5. When the desired datapoint is displayed, the value in the 'File Mask' field can be changed. This value is a bitmask, with the 'smaller' bits representing lower file numbers. The bit values are shown in Table 1. To calculate the appropriate value, add the numbers for each file the current datapoint should go to. Since the current datapoint is intended to go to the first and second files, the correct value should be 3, as 1 (file 1) + 2 (file 2) = 3.

Position/File Number	8	7	6	5	4	3	2	1
Value	128	64	32	16	8	4	2	1

Table 1- Values to use for File Mask.

When all appropriate datapoints have been modified, this use case is complete. If there is a large number of datapoints to change, it might be easier to modify the configuration file directly. The configuration file is described in Section 4.

## 4. Configuration Files

The configuration files used by the BDS contain information needed to specify the datapoint list, the network configuration, and other settings used by the BDS. The files are saved in an XML format, as shown in Figure 17. The elements of the configuration file are explained below.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<bdsDatapointList>
  <skipImportCheck/>
  <del />
  <dev MAC="10" rpm="5" routerMAC="00AA2A006367" label="VAV Zone 6" net="21" id="101" >
    <point type="0" id="1" label="bdp_1" tag="TAG0000" useDPL="1" unitID="56" maxVal="100" minVal="0" />
    <point type="0" id="2" label="chwr_1" tag="TAG0001" useDPL="1" units="degrees-Fahrenheit" unitID="64"
maxVal="96" minVal="45" /> </dev>
  <dev MAC="00BBB51422A" id="1" label="FD 1 " net="0" IP="192.168.2.127" />
  <dev MAC="01" id="226151" label="FD 11 " net="2263" routerMAC="00AAAA20F7D2" routerIP="192.168.1.123" />
  <dev MAC="11" rpm="5" routerMAC="00AA2A006367" label="VAV Zone 6" net="21" id="102" >
    <point type="0" id="1" label="2 ZA Temp" tag="TAG0001" />
    <point type="0" id="2" label="2 SA Temp" tag="TAG0002" />
    <point type="0" id="3" label="2 SA Flow CFM" tag="TAG0003" />
    <point type="1" id="0" label="2 MDMP" tag="TAG0016" />
    <point type="1" id="1" label="2 RH Coil" tag="TAG0017" />
  </dev>
  <var label="var1" value="12.3" />
  <alias label="alias1" dpname="bdp_1" />
  <dir>c:\</dir>
  <file>
    <name>bds.csv</name>
    <timer>10</timer>
  </file>
</bdsDatapointList>
```

Figure 17- Sample configuration file

**bdsDatapointList**- Root node for the file. All configuration elements must be within this node.

**dev**- This node describes a device on the network. The data relevant to network configuration are saved as XML properties. The dev node uses the following properties:

**MAC**- The MAC address of the device. This is needed for BACnet/IP, Ethernet, or MS/TP based networks.

**routerMAC**- The MAC address of the router, if present. This may be omitted if no router is used to connect to the device.

**net**- The BACnet network number of the network to which the device is directly connected.

**IP**- The IP address of the device, if BACnet/IP is used.

**routerIP**- The IP address of the router, if BACnet/IP is used. This may be omitted if no router is used to connect to the device, or if the router does not use BACnet/IP.

**label**- A text string used to describe the device.

**id**- The instance number of the Object\_Identifier on the Device object.

**IPport**- The IP port used by the device. Defaults to 0xBAC0 if absent.

**routerIPport**- The IP port used by the router. Defaults to 0xBAC0 if absent.

**rpm**- The count of the number of Property\_Identifiers requested in a ReadPropertyMultiple service request to this device. This determines the size of the packets sent when requesting data. If this value is too large, and the controller does not support segmentation, a controller will not be able to send a response to the request, and will instead generate an error message.

**point**- This node describes a BACnet object on the device. This is often also an element of the datapoint list. The properties which may be used in a point node are:

**type**- The type of BACnet object, using the enumerated value. Valid types that can be added to the datapoint list are 0-5. For example, for an Analog Output (AO) object, the value of type would be 1.

**id**- The instance number of the current BACnet object. For example, for the Analog Input 5 (AI05) object, the value of id would be 5.

**mask**- A bitmask determining to which output files the current datapoint is saved. Setting the least significant byte (LSB) indicates the first file. For example, a value of 3 would save to the first two output files. If omitted, this will default to a value of 1, which saves only to the first output file.

**label**- A text label used to describe the datapoint.

**unitID**- The value of the BACnet enumeration of the unit type. This is retrieved from the device.

**unit**- A text string indicating the units used for this datapoint. The string is the value of the enumeration indicated by unitID.

**maxVal**- The maximum value allowed for this BACnet object on the device.

**minVal**- The minimum value allowed for this BACnet object on the device.

**useDPL**- A value of 0 indicates that this point should be excluded from the datapoint list. Any other value indicates it should be included. Points will only be included if their type is in the range 0-5. Note that this defaults to 0 if not present. Use the “skipImportCheck” setting to include all appropriate points by default.

**tag**- An optional text string indicating the order of datapoints. This may be useful for debugging problems with the configuration of the BDS.

**var**- Creates a variable with an assigned value. Intended to be accessed by scripts.

**alias**- Creates an alias to an existing datapoint. Intended to be used by scripts.

**dir**- A text string indicating the directory to save files in, i.e., “c:\BDS”

**file**- This node describes an output data file. Note that there may be multiple output files designated, each with its own file node. The file nodes also have properties:

**name**- A text string containing the name of this output data file.

**timer**- The interval, in seconds, at which data is to be saved to this file

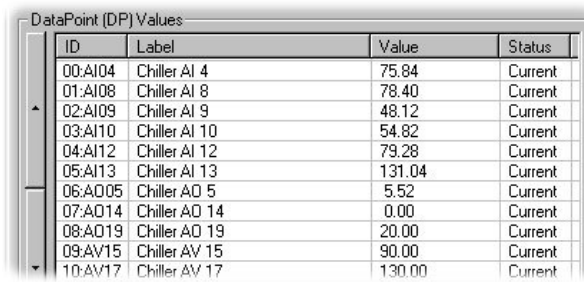
**del**- Instructs the BDS to delete the current datapoint list, device list, and output file list before reading in the current file.

**skipImportCheck**- Overrides the 'useDPL' parameter of point nodes to include all appropriate points.

The order of the nodes is not important as long as the proper hierarchy is maintained. Note that the exceptions to this are that the datapoint list will be constructed in the order that point nodes are found in the file, and the names of output files will be ordered similarly. The dir, file, del, or skipImportCheck nodes can be placed before or after the dev nodes, in any order.

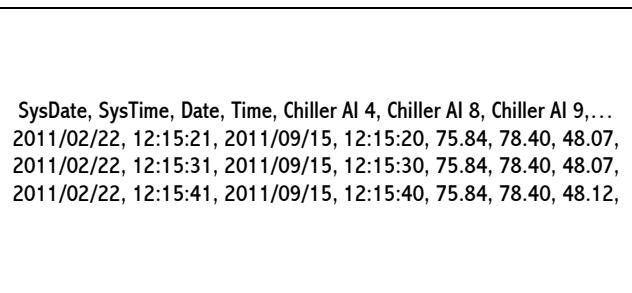
## 5. Details of data file operations

The data collected by the BDS can be saved into several files. The file name, directory, list of datapoints, and the interval to save the datapoints are given in the configuration file. The file directory can be changed at runtime by entering a new directory name in the Log File Location box. The interval at which data are saved for each file can also be changed at runtime by entering a new interval, in seconds, in the appropriate Data File Timer box. The datapoints are saved to the output files in the order they are listed in the datapoint list, and are separated by commas (CSV format). Before the first set of data is written to the datafiles, the label parameters from the elements in the datapoint list are written to the files to serve as column headings. Each subsequent line will contain a timestamp, followed by the data in the order designated. The timestamp consists of the date and time from the computer clock, formatted as “YYYY/MM/DD, hh:mm:ss”, followed by the date and time from the device indicated by the first element in the datapoint list, and then the data. An example is shown in Figure 18 and Figure 19.



ID	Label	Value	Status
00:AI04	Chiller AI 4	75.84	Current
01:AI08	Chiller AI 8	78.40	Current
02:AI09	Chiller AI 9	48.12	Current
03:AI10	Chiller AI 10	54.82	Current
04:AI12	Chiller AI 12	79.28	Current
05:AI13	Chiller AI 13	131.04	Current
06:AQ05	Chiller AQ 5	5.52	Current
07:AQ14	Chiller AQ 14	0.00	Current
08:AQ19	Chiller AQ 19	20.00	Current
09:AV15	Chiller AV 15	90.00	Current
10:AV17	Chiller AV 17	130.00	Current

Figure 18- Datapoints configured in BDS.



```

SysDate, SysTime, Date, Time, Chiller AI 4, Chiller AI 8, Chiller AI 9, ...
2011/02/22, 12:15:21, 2011/09/15, 12:15:20, 75.84, 78.40, 48.07,
2011/02/22, 12:15:31, 2011/09/15, 12:15:30, 75.84, 78.40, 48.07,
2011/02/22, 12:15:41, 2011/09/15, 12:15:40, 75.84, 78.40, 48.12,

```

Figure 19- The same datapoints saved to a CSV formatted file.

There may be up to 8 data files specified. Old data files are not deleted by the BDS, and new data will be added to them until they are moved or deleted by the user.

## **6. References**

- [1] Bushby, Steven T., et al., 2009, “The Virtual Cybernetic Building Testbed-A Building Emulator,” ASHRAE Transactions.
- [2] Bushby, Steven T, et al., 2001, “Using the Virtual Cybernetic Building Testbed and FDD Test Shell for FDD Tool Development”, NISTIR 6818.
- [3] Galler, Michael A. 2008, “Using the BACnet Communications DLL v1.0”, NIST Technical Note 1607, p. 29.
- [4] ASHRAE Guideline 20-2010, 2010, “Documenting HVAC&R Work Processes and Data Exchange Requirements,” ASHRAE.