Task Report for the

Energy Efficient and Affordable Small Commercial and Residential Buildings Research Program

A Public Interest Energy Research Program
Sponsored by the California Energy Commission

Project 2.7 – Enabling Tools

Task 2.7.2a –Use of the BACnet Data Source in the FDD Test Shell for Testing of FDD Tools in Real Buildings

Michael A. Galler, Mechanical Systems and Controls Group

Steven T. Bushby, Mechanical Systems and Controls Group

August 20, 2002

Prepared for
Architectural Energy Corporation

National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

Introduction

The Fault Detection and Diagnostics (FDD) Test Shell was developed as part of IEA Annex 34. It is a tool that can import building system data in a variety of file formats and make that data available to FDD tools through a Dynamic Data Exchange (DDE) interface. The idea was to make a tool that could speed the development of FDD tools by eliminating the complexity often involved in obtaining and processing trend data from various commercial control systems or output from simulation programs. The BACnet Data Source (BDS) extends the FDD Test Shell capabilities by adding a component that can exchange messages with BACnet controllers as a way to collect the data needed by the FDD tool. The data can be made available in real time or stored in a file for later retrieval and analysis. The interface between the Test Shell and the FDD tool being tested remains the same, thus the details of obtaining data from a BACnet control system are abstracted away in the same way that the details of processing simulation or trend data from other sources was abstracted away in the original FDD Test Shell.

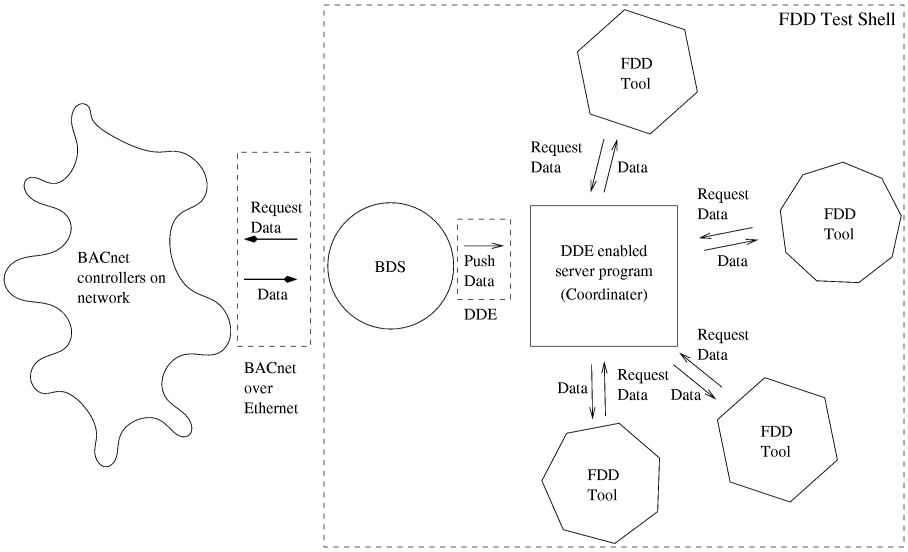

Figure 1- Diagram of BDS functionality

Figure 1 shows how the components of the FDD Test Shell interact. The BDS is designed to run on a personal computer with an Ethernet network interface card. It assumes that at least one part of the BACnet system uses an Ethernet local area network (LAN) that can serve as the connection point to the entire system. Data from BACnet controllers that reside on other BACnet subnetworks are obtained by communicating through the BACnet routers that are part of the control system. In this project the BDS was used with BACnet controllers in the Virtual Cybernetic Building Testbed. In principle, it could also be used in any real building that has a BACnet control system with an Ethernet LAN.

Description

The FDD Test Shell is comprised of three main parts: a data source, a data coordinator, and one or more FDD logic components. The data source can be one of several input file types or the BDS. The data source provides data to the coordinator at configurable time intervals. The coordinator alerts the FDD logic components when new data is available and they access the data through the DDE interface. The FDD logic components are the tools that are being evaluated. If multiple logic components are used they may process the same data with different rules or techniques, or they may process data from different HVAC controllers. For this project there was only one logic component used, called the AHU Performance Assessment Rules (APAR). APAR was implemented as a MATLAB module.

The BDS works by sending BACnet ReadPropertyMultiple requests to the building controllers. The requests currently can be for the Present_Value property of Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, and Binary Value objects, or for the Local_Date or Local_Time properties of a Device object. The data points to be retrieved by the BDS are determined by reading a configuration file at startup. The configuration file is divided into two sections. The first section lists the data points to be retrieved and the second section lists the location and names of files in which the data are to be stored. A useful feature of BDS is that subsets of the data points can be saved in separate output files, and any single data point can be saved in more than one output file. If a data point expected by the FDD Test Shell is unavailable, a placeholder or dummy value may be included in the data file to fill in for the missing value. A value of –999.9 will be stored for the missing value.

Configuration File Format

A sample configuration file is shown in Figure 2. In the first section of the configuration file, each line represents one data point that will be read by the BDS.

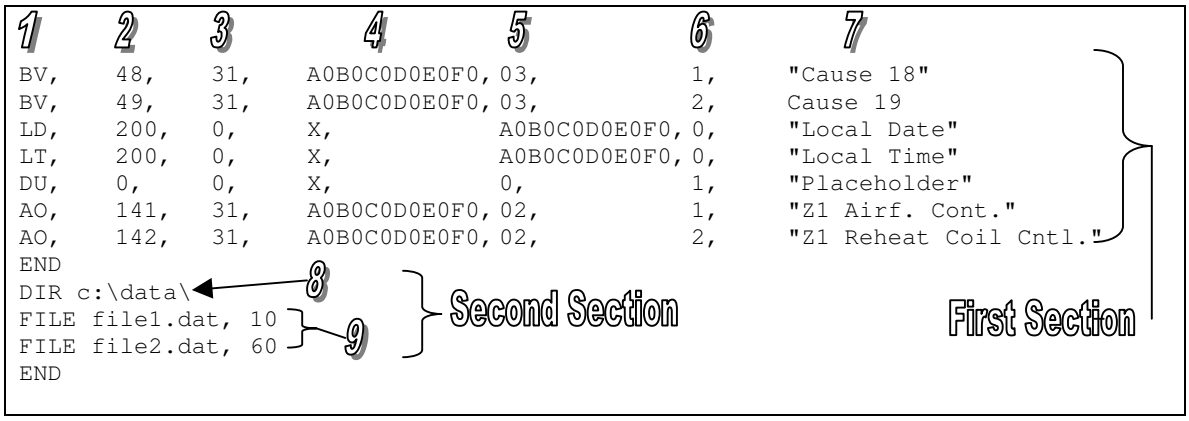

Figure 2 – A sample configuration file

The components of a line are:

1. Object-Property Listing: This entry must be selected from Table 1.

Entry	Object Type
AI	Analog Input
AO	Analog Output
AV	Analog Value
BI	Binary Input
BO	Binary Output
BV	Binary Value
LD	Device, Local_ Date
LT	Device, Local_Time
DU	Placeholder Value

Table 1- Object Types for configuration file

In BACnet, Local_Date and Local_Time are properties of the Device object, but for the purposes of BDS they are designated as separate entries above. For the other BACnet objects listed, the Present_Value property is used.

2. Object Instance: This is the instance number of the object. For example, if you want to read Analog Input number 4, this entry would be a 4.

3. Controller Network: This is the destination network address (DNET) of the controller containing the requested object. If the controller is on the same network as the computer running BDS, set this value to 0.

4. Router MAC Address: Enter the MAC address in hex of the first router to the controller network. If a router MAC address is not needed, enter an X instead.

5. Controller Address: Enter the MAC address of the controller here, in hex.

6. File Control Mask: This field controls where the data values are stored. A data value can be designated to be sent to the coordinator (or another DDE speaking program), or to any of up to eight separate files. The files are designated in the next section of the configuration file. This field operates as a bitmask, with the fields designated in Table 2. The value for the file control mask is the sum of the desired values in Table 2. It is not necessary to use or designate every file listed in Table 2. If the file control mask for a data point is set to save to a file that is not defined in section 2 (or a file which is later removed from the configuration file) the data point will still be saved to other files set by the file control mask. It should be noted that the Local_Date and Local_Time values, if designated, will automatically be saved to every data file, and the file control mask for those data points should be 0 to avoid duplication.

Value	Action
1	Sent through DDE, and saved in file 1
2	Saved in file 2
4	Saved in file 3
8	Saved in file 4
16	Saved in file 5
32	Saved in file 6
64	Saved in file 7
128	Saved in file 8

Table 2- File control mask values

Example: If a data point is to be saved in files 1, 3, and 6 then the file control mask for that data point would be 1 + 4 + 32 = 37.

7. Text Label: A descriptive label may be added to provide information about the data point. This label is used as a column heading for files in which values for this data point are stored. The label may be up to 40 characters long. Quotation marks surrounding the text are not necessary and will be removed from the text.

A line starting with the keyword END denotes the end of the first section. There may be up to 200 data points listed in the first section. For placeholder or dummy data points, only the Object-Property Listing, File Control Mask, and Text Label fields are used. The other fields must be present, but may contain any in-range values.

The second section of the configuration file describes the data files saved by BDS. There can be up to 8 files saved, each containing a subset of the data points. Data points can be saved to multiple files. There are two settings in the second section:

8. The first setting contains the directory in which the data files are to be stored. This setting is optional. If it is not included the default save directory is c:\. This setting is specified by starting the first line in the second section of the configuration file with the keyword DIR, followed by the directory in which the data files are to be saved, e.g., C:\data\. This setting is shown in the sample configuration file in Figure 2. If the directory entered is not usable, the directory will default to c:\.

9. The second setting describes an individual data file. This setting is specified by starting a line with the keyword FILE, followed by white space, then the filename with extension, a comma, and a number representing the interval in seconds to update the file. This number should be in the range from 10 to 3600. The first file will contain all the data from data points with a file control mask containing 1, the second file will contain all the data from data points with a file control mask containing 2, and so on up to 8 files.

A line starting with the keyword END also denotes the end of the second section. Any text placed after the end of section 2 will be ignored by BDS, allowing comments or other information to be placed there.

Details of data files

The data collected by the BDS can be saved into several files. The file name, directory, list of data points, and the interval to save the data points 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 save interval for each file can also be changed at runtime by entering a new interval, in seconds, in the appropriate Data File Timer box. The data values are saved to the data files in the order they are listed in section 1 of the configuration file, and are separated by commas (CSV format). Before any data is written to the data files, the label fields are written to the files to serve as column headings. The data that are saved in the first file (set by file control mask) are also sent to the DDE component at each interval, again in the order that they are listed in section 1 of the configuration file. There may be up to 8 data files specified. The data points that are listed for the first file are also sent to the DDE component. Old data files are not deleted by the BDS, and new data will be added to them until the user moves them.

Using the BDS

The first step is to determine the list of data points that will be recorded. This will usually be dictated by the requirements of the FDD tools being tested, although additional data points that are not available through the DDE interface may be recorded by the BDS. There may be up to 200 data points total, from multiple controllers on different networks. The data points may contain data from Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, or Binary Value object Present_Value properties, or the Local_Time or Local_Date from the Device object. It is suggested that data points from the same controller be grouped together to take advantage of the ReadPropertyMultiple capability of the BDS. If the FDD Test Shell is expecting values that are not available from the controllers, a placeholder or dummy value may be included in the configuration file with the file control mask set to include the first data file. The second step will be to determine the files to be saved by BDS and which data points will be saved to which files. Once that is done, the settings file can be created. The settings file must be in text format. The file name must only contain characters allowed for a file name. There are no other limitations on the name or file extension given to the settings file. The next step is to start the BDS. Figure 3 shows BDS after it is opened. After BDS is opened, the name and location of the settings file must be entered in the box in the 'Select Level to Monitor' area, as shown in Figure 4. The default file name is c:\custom.bif. The path and filename together must less than 80 characters long. When the full path and filename are entered, click on the 'Custom Floor' button to have the BDS read in the settings file. If there are any errors in the settings file, an error message will appear in the status window. After the settings file is read in, the Datapoint Values area will show the data points from the settings file and the Data File Timers area will show a box for each data file containing the update interval for that file. The update interval for a data file may be changed to any number of seconds in the range 10 to 3600 by typing a new number in the box for that data file. Because the first data file listed in the settings file determines how often the data is updated, it is suggested that subsequent data files have an update interval equal to or larger than the update interval for the first data file to avoid over sampling the data.

After the settings file has been read, the next step is to initialize the BACnet interface. This is done by pressing the button labeled "Init BACnet" in the control button area of the BDS

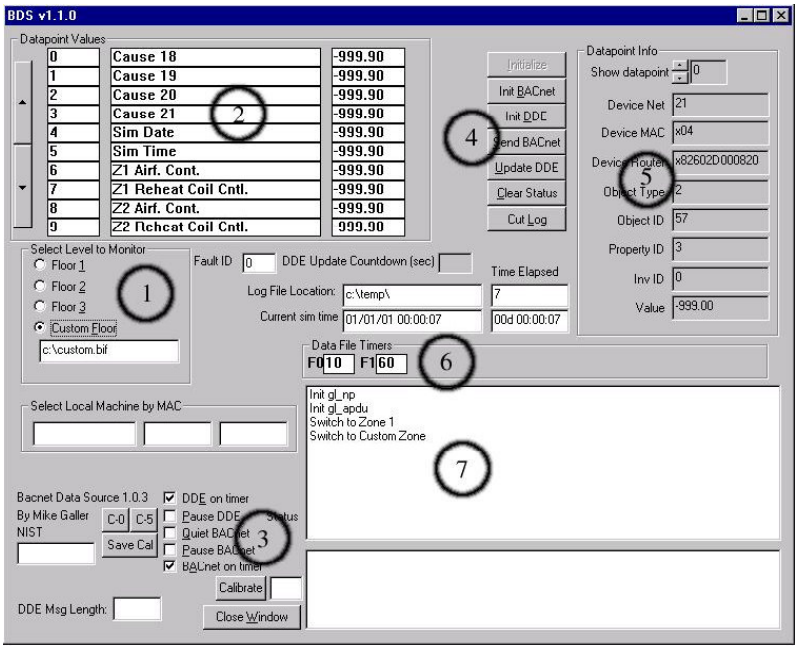

Figure 3: View of the BDS when opened.

1: Click Custom Floor to read in settings file
2: The data points will be displayed here
3: Settings for BDS
4: Controls for BDS
5: Details on data points viewed here
6: Details on data file timers viewed here
7: Status window displays events and error messages

interface. If there are any errors, an error message will appear in the status window. This step will also start the timers used to control the data files. If data is not being sent to the Coordinator (i.e. only saving data to file), then this is the last step until data collection is finished.

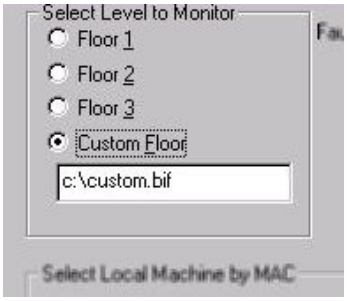

Figure 4- Enter the name and location of the settings file

If data is being sent to the Coordinator, then the DDE interface will also need to be initialized. The Coordinator must be started, then press the button labeled "Init DDE" which is right under the "Init BACnet" button. This will establish a connection to the Coordinator. At this point, data

collection and passing will have begun, and no further action is needed until data collection is finished.

Format for output data files

The format for the output data files is consistent across all data files. The variables are saved to the data files in the order they are specified in the configuration file. Each file will start with a row of column headings, consisting of the data label for each data point designated to be saved into that file. The second line will contain the string "data = [" which is intended to assist with importing the data files into Matlab. The data starts on the third line. The format for each line begins with the date as read from a controller in YYMMDD format, then the time as read from a

```
Date, Time, Cycle, Occupancy, Spy. Temp Setpoint, Spy. Air Temp
data = [
010201, 0630, 38.00, 00.00, 15.56, 12.35
010201, 0631, 96.00, 00.00, 15.56, 12.19
010201, 0632, 153.00, 01.00, 16.11, 10.98
]
```

Figure X- A sample data file, starting 6:30 A.M., Feb. 1, 2001

controller in hhmm format, then the cycle counter, which records the number of seconds elapsed since data collection began. These are followed by the data points listed in the configuration file, with every variable separated by a comma. If there is no time and/or date data point listed in the configuration file, then the current time and date on the host computer will be used. The last line of the data file will contain a right bracket, "]", again to assist with Matlab. The data files will be renamed with a time/date stamp when the date being saved to the data file changes, or when the 'Cut Log' control button is pressed. The format for the renamed log file is "name-YYMMDD-hhmm-DX.ext", where YYMMDD is the year, month, and day of month on the host computer, hhmm is the hours and minutes on the host computer, X is the number of days the BDS has been running, and where the original file name was "name.ext"

To pause data collection, click on the "Pause BACnet" checkbox. This will halt all BACnet communication from the BDS until the checkbox is cleared. Also, data will not be written to any data files and no data will be sent to the Coordinator. To halt all data transfer to the Coordinator, but still allow BACnet communication and collect data in the files, click on the "Pause DDE" checkbox.

To end data collection, click on the "Close Window" button. This will also rename the data files as described above.