

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 2.7.3 – Development of a BACnet Interface
for the Whole-Building Diagnostician
(WBD)**

Mike A. Galler, National Institute of Standards and Technology
Srinivas Katipamula, Battelle Pacific Northwest Division

August 2002

Prepared for
Architectural Energy Corporation

Battelle Pacific Northwest Division
Richland, Washington 99352

THIS REPORT WAS PREPARED AS A RESULT OF WORK SPONSORED BY THE CALIFORNIA ENERGY COMMISSION (COMMISSION). IT DOES NOT NECESSARILY REPRESENT THE VIEWS OF THE COMMISSION, ITS EMPLOYEES, OR THE STATE OF CALIFORNIA. THE COMMISSION, THE STATE OF CALIFORNIA, ITS EMPLOYEES, CONTRACTORS, AND SUBCONTRACTORS MAKE NO WARRANTY, EXPRESS OR IMPLIED, AND ASSUME NO LEGAL LIABILITY FOR THE INFORMATION IN THIS REPORT; NOR DOES ANY PARTY REPRESENT THAT THE USE OF THIS INFORMATION WILL NOT INFRINGE UPON PRIVATELY OWNED RIGHTS. THIS REPORT HAS NOT BEEN APPROVED OR DISAPPROVED BY THE COMMISSION NOR HAS THE COMMISSION PASSED UPON THE ACCURACY OR ADEQUACY OF THE INFORMATION IN THIS REPORT.

Introduction

The BACnet Data Source (BDS – this is generically referred to as BASBACnet DLL) was developed to be used as a component of the Whole Building Diagnostician (WBD), in conjunction with work on the Virtual Cybernetic Building Testbed (VCBT) (reference 1). The VCBT uses real BACnet enabled mechanical systems and controllers coupled with computer simulations to emulate an entire 3 story building, with three zones per level. Each level has an AHU controller and 3 VAV controllers, with each VAV controller connected to a single zone. The AHU controllers on each level are from three different manufacturers.

The BDS or the BASBACnet provides a data link between the VCBT BACnet controllers/devices and the WBD database (Figure 1). BASLink (Building Automation System Link), a component of the WBD, retrieves data from the VCBT controllers through the BASBACnet and makes the data available to the WBD modules (OAE and WBE) in real-time. After receiving a request from the BASLink process, the BASBACnet retrieves data from the controllers/devices by communicating with them using the native BACnet protocol.

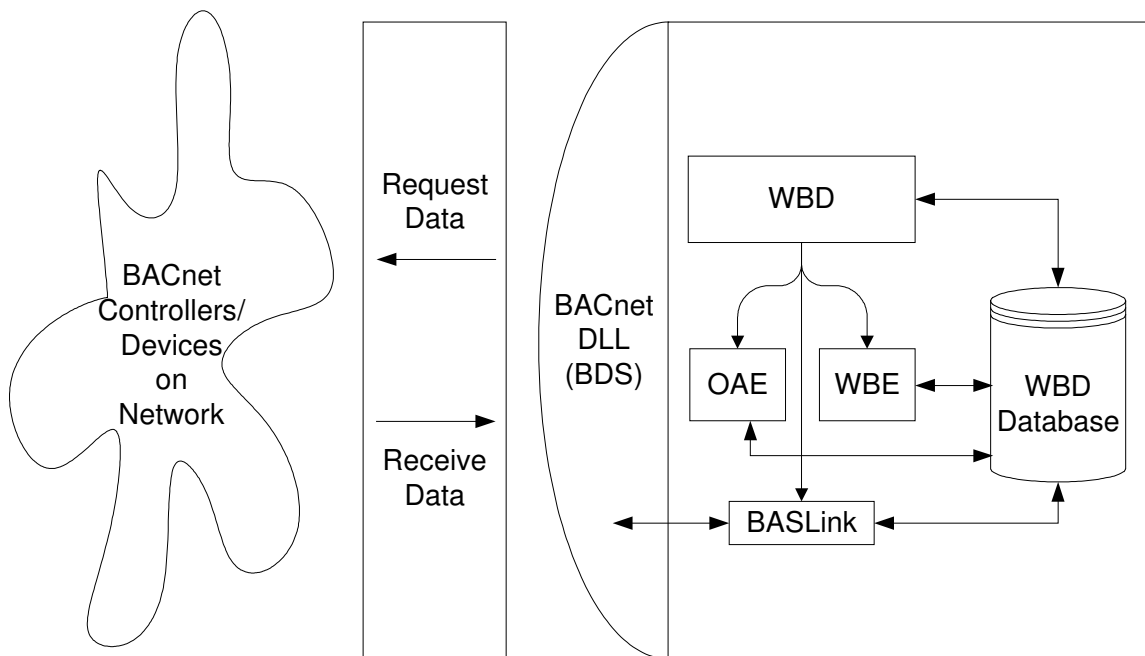


Figure 1 – Overview of the Data Exchange Process

The BASBACnet DLL is capable of retrieving values from BACnet objects on BACnet speaking controllers on an Ethernet network and passing them back to the WBD. While the BDS was developed in conjunction with the VCBT and the WBD, it can be used in any building with a computer connected to a BACnet network¹. The BASBACnet is capable of retrieving data from multiple controllers on different BACnet subnets, which is an improvement over previous VCBT WBD implementation.

¹ Currently, one of the BACnet support DLLs only runs in a Windows NT 4.0 environment; therefore the data can only be retrieved from by Windows NT 4.0 based computers. The WBD has no such limitations and runs on all Windows operating systems including Windows XP.

Description of the BASBACnet DLL

The BASBACnet DLL is capable of retrieving data from the BACnet speaking controllers/devices connected over an Ethernet network, and returning that data to the WBD database, or any other program that uses it. The BASBACnet DLL can currently handle the following requests for the Present-Value property of Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, and Binary Value objects only.

Configuration

For the BASBACnet to work properly the network information of a BACnet controller or device is required: the BACnet network number, the controller MAC address, and the router MAC address. There are two ways to configure the network information 1) in an initialization file, or 2) with each data point. If all data that are being requested come from a single controller, the network information can be specified in the optional initialization file (wbdlink.ini), as shown in Figure 2. If the requested data spans different controllers that are on different networks (different subnets), then the network information must be associated with each data point.

```
# 1 controller MAC in hex
03

# 2 router MAC in hex, or X if none
0A0B0C0D0E0F

# 3 subnet is in decimal
31
```

Figure 2 – Sample Network Setting File

In addition to the network information of the controllers/devices, a list of data points for which the present value is needed must be listed in the WBD database. Each entry must contain the object type, the object id, and the requested property. The entry may optionally contain network information, which will override the default values for that data point (if an initializing file is specified), as shown in Figure 3. The fields of the entry are:

1. **Device:** The BACnet object instance – this is a required.
2. **Object:** The BACnet object type – this is a required. The possible values for this are: 0 for Analog Input, 1 for Analog Output, 2 for Analog Value, 3 for Binary Input, 4 for Binary Output, or 5 for Binary Value.
3. **Attribute:** The BACnet property (3 or 85 for present-value) – this is a required field.
4. **DMAC:** The MAC address of the controller, in hex – this is an optional field.
5. **RMAC:** The MAC address of the BACnet router, in hex, a value of “X” indicates local network – this is an optional field.
6. **DNET:** The network the controller is on, in decimal, use “0” for local network – this is an optional field.

Sample data entries with an initialization file:

(Device: 1, Object: 0, Attribute: 3) use default network info for DMAC, RMAC, and DNET

(Device: 1, Object: 4, Attribute: 3, DMAC: 03, RMAC: 010203040506, DNET: 20) override default network info for this entry

(Device: 1, Object: 3, Attribute: 3, DMAC: 05) override default DMAC, but use default RMAC and DNET

(Device: 2, Object: 1, Attribute: 3) uses default network info for DMAC, RMAC, and DNET from settings file

Sample data entries without an initialization file:

(Device: 1, Object: 0, Attribute: 3, DMAC: 03, RMAC: 0A0B0C0D0E0F, DNET: 20) set default network info - must be first

(Device: 2, Object: 2, Attribute: 3) use default network info for DMAC, RMAC, and DNET set in previous entry

(Device: 1, Object: 3, Attribute: 3, DMAC: 05) set new default DMAC, but use previous RMAC and DNET

(Device: 1, Object: 3, Attribute: 3, DNET: 30) set new default DNET, but use previous DMAC and RMAC

(Device: 2, Object: 0, Attribute: 3) uses DMAC=5, RMAC=0A0B0C0D0E0F, and DNET=30 as set in previous entries

Figure 3 – Sample Data Entries for Defining the Point Information

If the initialization file is not present, the network information must be present on the first data point entry in the database. That network information is then used as the default, until a data point entry with new network information is read in, after which the latest network information read in will be used as the default.

The entries shown in Figure 3 are entered in the column named **SourceID** within the table named **ChannelConfig** in the WBD database.

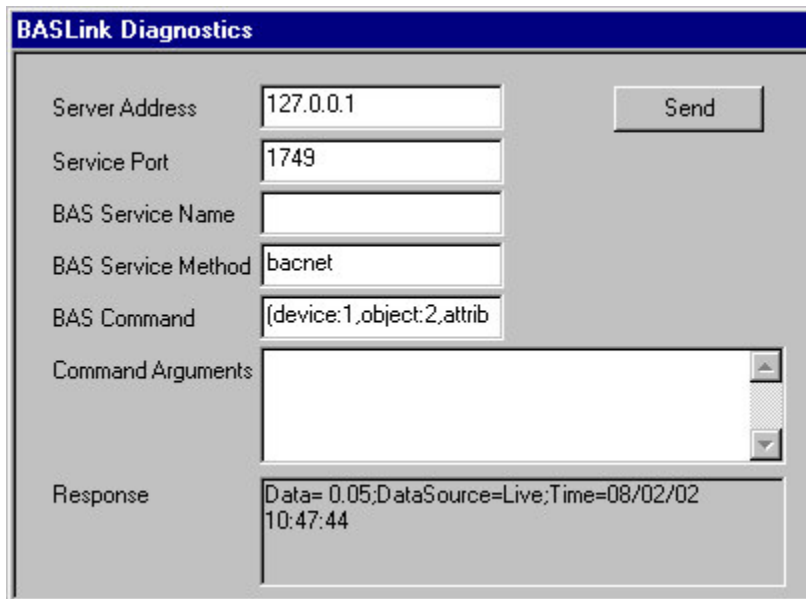
Process

The WBD diagnostic modules expect the data from the AHUs and the end-use consumption to be present in the WBD database. Therefore, these data tables have to be populated on a continuous basis using the BASLink component of the WBD. The BASLink module checks the **ChannelConfig** table in the WBD database for a list of data points and passes each data request to the BASBACnet DLL. The BASBACnet takes the request from the BASLink and sends a

BACnet ReadProperty requests to controllers/devices. The data points are uniquely identified using the device and object identification information provided as part of the configuration. The controller/device sends the requested data to the WBD database through the BASBACnet and BASLink modules. The data are continuously read and accumulated in a temporary table. At the end of each hour, the average values are computed and the data moved to relevant tables.

Testing

The BASBACnet DLL was tested using the WBD BASLink Diagnostic module. The BASLink Diagnostic module allows for testing the data collection process without having to configure the database. The relevant configuration information is entered directly in the relevant field as shown in Figure 4.



BASLink Diagnostics	
Server Address	127.0.0.1
Service Port	1749
BAS Service Name	
BAS Service Method	bacnet
BAS Command	(device:1,object:2,attrib
Command Arguments	
Response	Data= 0.05;DataSource=Live;Time=08/02/02 10:47:44

Figure 4 – BASLink Diagnostics Dialog Showing the Configuration Information and the Response from the BACnet Controllers

Future work

The future direction of the distributed version of BASBACnet will depend largely on the needs of the user base. Some anticipated areas for future development are:

1. Increased object/property support
2. Additional connectivity options

An area that is open for improvement is the transfer of data between the WBD and BASBACnet. The BASBACnet is designed to handle multiple requests for data, i.e. it can receive a new data request before previous data requests have been retrieved by the calling program. The WBD currently requires a data request to be returned before sending the next one, which takes more time. Implementing this in the WBD would probably be non-trivial, and is not necessary for the

current volume of data, but in the future if increased amounts of data are required this might be a useful change to make.