BACnet Today and the Smart Grid

Part One

BACnet® Explained

By H. Michael Newman, Fellow/Life Member ASHRAE

Let's face it: BACnet is pretty amazing. For one thing, this protocol (set of rules) for building automation and control system networking has been an ANSI/ASHRAE standard for 18 years. If it were a citizen, it would be old enough to vote (see sidebar "BACnet is 18!"). For another, it is continually being rejuvenated so that, in many respects, it is more youthful than the day it was born.

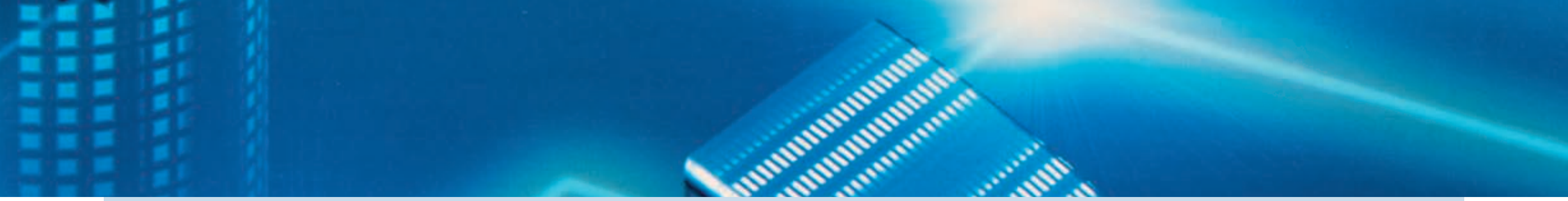
All of this is due to the fundamental solidity of the underlying concepts that have been in place since committee deliberations began, all the way back in 1987; and the incredible skill and dedication of the dozens of "BACneteers" who have worked tirelessly on the standard ever since. In this article, I will focus on BACnet's underlying concepts and how they all support BACnet's overriding purpose: interoperability. These concepts are the keys to BACnet's longevity and success. If you understand these, the 1,039-page standard won't seem so daunting!

Concept 1: BACnet uses an "object model" to represent the functioning of building automation and control systems (BACS). No matter what changes may occur over time to the rest of BACnet, its object model will probably survive pretty much as it is. This is where the functioning of BACS devices is "mapped" to a set of standardized collections of attributes. If you think back to where the industry was in 1987, every manufacturer had developed direct digital controllers and they all did more or less the same things. But each one did its internal work differently. The challenge for SPC 135P, the original BACnet committee, was to come up with a way to represent the functioning of these devices in a common, standard way. At the time, "object-oriented programming" was making its debut and we decided that we should take advantage, at least to some degree, of this object-oriented approach.

So an "object" in BACnet is just a collection of information that relates to the functioning of a particular BACS device. The collection of information is a set of "properties." Each property has an identifier, a data type (e.g., analog, binary, text, or whatever) and a conformance code that indicates whether it is required to be present or optional and whether it is read-only or can be written to, i.e., modified, by BACnet services (to be explained next). All object types are required to have Object_Identifier, Object_Name and Object_Type properties. One subtlety is worth mentioning. The standard defines "object types." Instances of object types in real devices are what are properly called "objects." Object types tell you what properties an object can have. The properties of an object, contain the actual values of the given properties as implemented. In any case, 135-2012 (BACnet-2012) now defines 54 object types, up from the original 18 in BACnet-1995. (See sidebar "BACnet Object Types").

About the Author

H. Michael Newman is the manager of the Building Automation and Control Systems Integration group at Cornell University. He was chair of the BACnet standard committee from 1987 to 2000.



BACnet is 18!

Believe it or not, BACnet has been a standard for 18 years. Here is a timeline of some of the most important events in its history:

- 1987—First meeting of SPC 135P, the "Energy Monitoring Control Systems Message Protocol" committee, in Nashville, Tennessee.
- 1990—Project name is officially changed to "BACnet."
- 1991—First public review begins.
- 1995—After three public reviews and 741 comments from 81 commenters in 11 countries, BACnet is finally approved and published as an ANSI/ASHRAE standard (135-1995). It is a petite document of a mere 501 pages.
- 1996—Standing SPC 135 is formed. BACnet becomes one of the first standards to be maintained under new "continuous maintenance" procedures.
- 1998—First BACnet Interest Group, BIG-Europe (BIG-EU), launches in Frankfurt, Germany. BACnet website, www.bacnet.org, goes online.
- 1999—BACnet/IP, the first addendum to BACnet-1995 is approved. BIG-North America (BIG-NA) is formed and BACnet becomes a Korean national standard. Engineering News-Record magazine selects BACnet as one of the "Top 125 Innovations of the Last 125 Years."
- 2000—The BACnet Manufacturers' Association (BMA) is formed and soon opens the BACnet Testing Laboratory. BACnet is translated into Chinese and Japanese. BMA sponsors first BACnet Interoperability Workshop ("PlugFest") at NIST with 12 organizations attending. (PlugFest-2012 had 49 teams.)
- 2001—The second consolidated version of the standard, BACnet-2001, is published.
- 2002—ASHRAE Journal publishes its first "BACnet Today" supplement.
- 2003—BACnet becomes an ISO standard, 16484-5. The BACnet testing standard, ANSI/ASHRAE 135.1, is published and is also accepted by the ISO as 16484-6. BACnet is accepted around the world.
- 2004—BACnet-2004 is published. BIG-EU publishes the first BACnet Europe Journal.
- 2005—BACnet International forms from the merger of BMA and BIG-NA
- 2006—ASHRAE issues the 200th"Vendor ID. (That number is now up to 693.)
- 2007—The BACnet testing standard, 135.1, is republished to stay aligned with BACnet itself.
- 2008—BACnet-2008 is published with a Web Services Interface and several new object types including Structured View, Load Control, Event Log, Trend Log Multiple, and Global Group.
- 2010—BACnet-2010 is published with ZigBee wireless data link, access control objects, NAT for BACnet/IP, and XML data formats.
- 2013—BACnet-2012 is published. It contains significant updates to BACnet's alarm and event mechanisms but the document is no longer petite: BACnet-2012 is 1,039 pages long!

This list is truly just the "tip of the iceberg." The number of interest groups, publications, major conferences and expositions, interoperability workshops, and other notable events is much too lengthy to include here. And this list doesn't even mention the hundreds of BACnet products that are now available or the thousands of projects in which they have been installed. Please visit www.bacnet.org and www.bacnetinternational.org for much more!

Objects solve the problem of representing the functions of a given BACnet device in a standard, network-visible way. Each object is uniquely identified within the device that maintains it by its Object_Identifier property. The combination of a unique internetwork-wide object identifier for the *device* and a unique object identifier within the device for each *object* allows the object to be unambiguously accessed or "addressed" over the network. Knowing the possible "property identifiers" for each object allows the value of the property to read and, possibly, written. The collection of a set of objects is called a "BACnet Device."

The object model was designed to be flexible and extensible. This allows new objects, both standard and proprietary, to be developed with ease, a concept that has stood the test of time!

Concept 2: BACnet defines a set of "services" to communicate about the functioning of BACnet Devices. Once the problem of how to represent a device's functionality was solved through the development of the object model, the BACnet committee had to figure out what BACS devices would want to say to each other and how they would want to say it. We settled on the concept of a "client-server" model where the client device sends a "service request" to the server device and the server responds with a "service response." The set of services that was developed are all aimed at facilitating interoperability. They allow alarms and events to be processed; files containing configuration and programming information to be exchanged; the object model to be exploited, i.e., properties to be read and written; the time to be synchronized; devices to learn about other devices and properties through "dynamic binding"; and many other things. There are currently 38 BACnet services (see Sidebar, "BACnet Services") that fall into five broad classes (*Table 1*).

BACnet Object Types

BACnet-2012 defines 54 object types that can be categorized as shown below. These definitions occupy about one quarter of the standard so once you learn how to understand any one of them, you can easily interpret the rest.

Basic Device Object Types
Device
Analog Input
Analog Output
Analog Value
Binary Input
Binary Output
Binary Value
Multi-state Input
Multi-state Output
Multi-state Value
File

Process-related Object Types
Averaging
Loop
Program

Control-related Object Types
Command
Load Control

Meter-related Object Types
Accumulator
Pulse Converter

Presentation-related Object Types
Group
Global Group
Structured View

Schedule-related Object Types
Calendar
Schedule

Notification-related Object Types
Event Enrollment
Notification Class
Notification Forwarder
Alert Enrollment

Logging Object Types
Event Log
Trend Log
Trend Log Multiple

Life Safety and Security Object Types
Life Safety Point
Life Safety Zone
Network Security

Physical Access Control Object Types
Access Zone
Access Point
Access Door
Access User
Access Rights
Access Credential
Credential Data Input

Simple Value Object Types
CharacterString Value
DateTime Value
Large Analog Value
BitString Value
OctetString Value
Time Value
Integer Value
Positive Integer Value
Date Value
DateTime Pattern Value
Time Pattern Value
Date Pattern Value

Lighting Control Object Types
Channel
Lighting Output

Concept 3: Encoded BACnet messages should be identical for all devices, large and small and the encoding should be flexible enough to allow for the easy addition of new capabilities. This idea was a bit radical at the time. Many people wanted a “stripped down” version of the protocol for small “dirtball” devices. We stood our ground, however, confident that the capabilities of small computing devices would inevitably improve—at least to the point of being able to process our protocol. Heck, my cell phone today has the computing power of a building full of 1990s-vintage controllers. The result is that implementers can use the same encoding/decoding software in any of their machines.

To ensure extensibility, we chose a “tagged” encoding method called *Abstract Syntax Notation One (ASN.1)*, ISO Standard 8824. ASN.1 uses numeric labels, i.e., tags, to explicitly identify each message parameter rather than depending on an implicit knowledge of the parameter’s position in a string of bytes. This means that message encoders and decoders can be written that are not dependent on a particular predefined, usually static, arrangement of a message’s contents, but instead can use the tags to figure out where one piece of data ends and another begins. Here is an example of a fictional ASN.1 “production” or structure:

```
Room-Conditions ::= SEQUENCE {
temperature  [0]  REAL OPTIONAL,
humidity     [1]  REAL OPTIONAL,
occupancy    [2] BOOLEAN OPTIONAL
}
```

So the contents of message could be:

Room-Conditions ::= {[0] 72.3, [2] TRUE}

meaning the temperature in the room is 72.3 and the room is occupied. Since each of the data elements is “OPTIONAL” its presence or absence is indicated by the presence or absence of the numeric tag 0, 1 or 2. In the example, the humidity value has not been provided so the [1] tag has been omitted, but it is clear that the “72.3” is the temperature, not the humidity, from its tag.

BACnet also describes how to convert the abstract representation of all its data types into 0s and 1s for conveyance on the communications medium and, of course, specifies how to address the encoded message to the correct device, how to check for transmission errors, and so on.

Concept 4: BACnet messages should be capable of being transported on whatever network technology makes sense. After the object model, services, and encoding techniques had been established, it was time to figure out how to get the bits and bytes from one machine to the next. We decided we needed a few different kinds of technology--ranging from fast and relatively expensive to slower and cheaper--to accommodate the range of field devices, their capabilities, and the requirements of a particular job.

We took the pragmatic approach and asked the vendors on the committee what kind of networks they were using. There would be no point in “standardizing” a technology that no one would use! We initially ended up with Ethernet; ARCNET; a protocol we developed ourselves called Master-Slave/Token-Passing (MS/TP) that runs over EIA-485 twisted pair wiring; a point-to-point protocol designed for dial-up telephone communi-

Service Class	Purpose	Number of Services
Alarm and Event	Services used to manage communication related to alarms and events.	11
File Access	Services used to access and manipulate files.	2
Object Access	Services used to access and manipulate the properties of BACnet objects.	10
Remote Device Management	Services used for a variety of miscellaneous, but important, functions such as time synchronization, starting and stopping communication, reinitializing processes, transferring proprietary messages and dynamic binding.	12
Virtual Terminal	Services that are now rarely used because of the advent of the web. These services provided for the establishment of the "bidirectional exchange of character-oriented data," according to the standard, for use by an operator interface program.	3

Table 1: All 38 BACnet services fall into one of these 5 broad "Service Classes."

B-AWS	B-OWS	B-OD	B-BC	B-AAC	B-ASC	B-SA	B-SS
DS-RP-A,B	DS-RP-A,B	DS-RP-A,B	DS-RP-A,B	DS-RP-B	DS-RP-B	DS-RP-B	DS-RP-B
DS-RPM-A	DS-RPM-A		DS-RPM-A,B	DS-RPM-B			
DS-WP-A	DS-WP-A	DS-WP-A	DS-WP-A,B	DS-WP-B	DS-WP-B	DS-WP-B	
DS-WPM-A	DS-WPM-A		DS-WPM-B	DS-WPM-B			
DS-AV-A	DS-V-A	DS-V-A					
DS-AM-A	DS-M-A	DS-M-A					

Table 2: Required BIBBs for the Data Sharing IA. The eight Device Profiles range from B-SS for "BACnet-Smart Sensor" to B-AWS for "BACnet-Advanced Workstation."

cations; and a specification for using Echelon Corporation's LonTalk.

The first addendum to BACnet-1995, BACnet/IP, added the ability to use the Internet Protocol and just a few years ago we added a specification for a wireless protocol called ZigBee. The standard also defines, in detail, how all of these network types can be combined in a seamless way to form a "BACnet Internetwork." So today there are a total of seven technologies to choose from for any given application and, if some new kind of network comes along, BACnet is perfectly positioned to take advantage of it, another example of why BACnet has been as successful as it has been!

Concept 5: BACnet needs to describe how to achieve interoperability in a standard way. Up until now, what I have described is just a large collection of capabilities--objects, services, encoding, and network types—from which choices must be made when implementing a real system. These capabilities can be thought of as tools in a toolbox and, like real tools, different ones can be chosen to accomplish more or less the same thing. Some tools make the job easier, some require more skill to use, etc.

Since interoperability is really what BACnet is all about, the standard provides guidance on how to make "tool selections" that will provide the greatest probability that equipment from different suppliers will work smoothly together for a particular function.

There are three pieces to the puzzle. The first is what the standard calls "Interoperability Areas" or IAs. There are five IAs: Data Sharing, Alarm and Event Management, Scheduling, Trending, and Device and Network Management. According to the standard, "*Each IA implies a set of capabilities. Each capability, in turn, requires that specific elements of BACnet be implemented in a particular device to enable interoperability in a known and predictable manner with a minimum of field engineering. The selection of which BACnet elements are required for a particular type of device is indicated in the device profiles presented in Annex L.*" Clause 22 of the standard provides a general overview of what each IA is intended to accomplish.

The second is the definition of "BACnet Interoperability Building Blocks" or "BIBBs" that are related to each IA. BIBBs are collections of one or more BACnet services prescribed in terms of an "A" device and a "B" device. The "A" device is generally the user of data (client), and the "B" device is generally the provider of the data (server). Each BIBB specifies whether a supporting device is expected to initiate or execute a particular service.

BIBBs may also be predicated on the support of certain BACnet objects or properties and may place constraints on the allowable values of specific properties or service parameters. BIBB names are of the form "IA Abbreviation"-"Service Abbreviation"-"A or B". As an example, "DS-RP-A" means "Data Sharing"- "Read-Property service"-"A-side device". In the standard, this BIBB looks like this:

BIBB - Data Sharing - ReadProperty-A (DS-RP-A)

The A device is a user of data from device B.

BACnet Service	*Initiate*	*Execute*
ReadProperty	*x*	

So a device that supports DS-RP-A is expected to be able to generate a ReadProperty service request. Pretty simple. A B-side device, as you can probably guess, is expected to be able to respond to a ReadProperty request. BACnet-2012 contains 111 BIBBs, all of them defined in Annex K.

The third part of the interoperability puzzle is the definition of a set of BACnet Device Profiles. There are currently eight of them:

- BACnet Operator Workstation (B-OWS)
- BACnet Advanced Operator Workstation (B-AWS)
- BACnet Operator Display (B-OD)
- BACnet Building Controller (B-BC)
- BACnet Advanced Application Controller (B-AAC)
- BACnet Application Specific Controller (B-ASC)
- BACnet Smart Actuator (B-SA)
- BACnet Smart Sensor (B-SS)

The description of each profile in Annex L starts with a thumbnail explanation of the device and is followed by the expected capabilities in each of the five IAs. Here, for example, is the description of a device of moderate complexity, a B-AAC:

BACnet Services

BACnet-2012 defines 38 services that are the basis for all inter-device messages.

Alarm and Event Services
AcknowledgeAlarm
ConfirmedCOVNotification
UnconfirmedCOVNotification
ConfirmedEventNotification
UnconfirmedEventNotification
GetAlarmSummary
GetEnrollmentSummary
GetEventInformation
LifeSafetyOperation
SubscribeCOV
SubscribeCOVProperty

File Access Services
AtomicReadFile
AtomicWriteFile

Object Access Services
AddListElement
RemoveListElement
CreateObject
DeleteObject
ReadProperty
ReadPropertyMultiple
ReadRange
WriteProperty
WritePropertyMultiple
WriteGroup

Remote Device Management Services
DeviceCommunicationControl
ConfirmedPrivateTransfer
UnconfirmedPrivateTransfer
ReinitializeDevice
ConfirmedTextMessage
UnconfirmedTextMessage
TimeSynchronization
UTCTimeSynchronization
Who-Has
I-Have
Who-Is
I-Am

Virtual Terminal Services
VT-Open
VT-Close
VT-Data

BACnet Advanced Application Controller (B-AAC)

A B-AAC is a control device with limited resources relative to a B-BC. It may be intended for specific applications and supports some degree of programmability.

Data Sharing

- Ability to provide values for any of its BACnet objects upon request
- Ability to allow modification of some or all of its BACnet objects by another BACnet device

Alarm and Event Management

- Generation of limited alarm and event notifications and the ability to direct them to recipients
- Tracking acknowledgments of alarms from human operators
- Adjustment of alarm parameters

Scheduling

- Ability to schedule actions in the local device based on date and time

Trending

- No requirement

Device and Network Management

- Ability to respond to queries about its status
- Ability to respond to requests for information about any of its objects
- Ability to respond to communication control messages
- Ability to synchronize its internal clock upon request
- Ability to perform re-initialization upon request

The actual profiles that tie all these elements together have tables for each of the IAs and show the required BIBBs for each type of device. So, for example, for the Data Sharing IA, *Table 2* lists the required BIBBs.

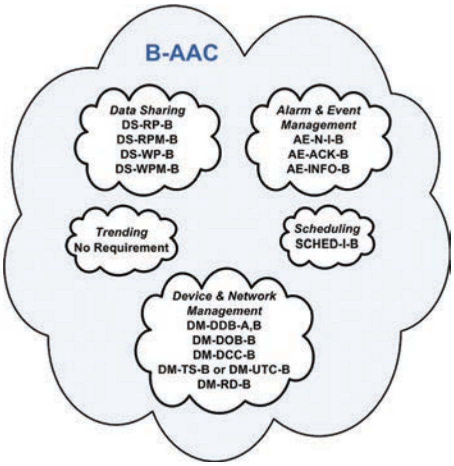


Figure 1: Each Device Profile specifies of set of BIBBs for each Interoperability Area as shown here for a *BACnet—Advanced Application Controller* (B-AAC).

So for a device claiming to be a B-AAC, it must support the B-side BIBBs for ReadProperty and ReadPropertyMultiple (DS-RP-B and DS-RPM-B) as well as the B-side BIBBs for WriteProperty and WritePropertyMultiple (DS-WP-B and DS-WPM-B) (*Figure 1*).

To see how all of this works, I urge you to go to www.bacnetinternational.org. Under the "Products" tab select "BTL Listed Products" There you will find a table with the manufacturers listed vertically and the Device Profiles. Selecting a manufacturer link leads you to a list of all of that manufacturer's products, arranged by Device Profile. If you select one of the Device Profile links, you can see all of the manufacturers who have listed products (i.e., BTL satisfactorily tested products) meeting that profile. These products should all work together well!

In "BACnet Explained," Part 2, Grant Wichenko describes some of the new capabilities of our ever-evolving standard. ■