Proceedings of the 1999 IEEE
International Conference on Control Applications
Kohala Coast-Island of Hawai'i, Hawai'i, USA • August 22-27, 1999

The Open Protocol Standard for Computerized Building Systems: BACnet

Larry K. Haakenstad
Alerton Technologies, Inc.
6670 185th Ave. NE
Redmond, WA 98033

What Is BACnet?

BACnet is the term commonly used to refer to the ANSI/ASHRAE Standard 135-1995, adopted and supported by the American National Standards Instituite (ANSI) and the American Society of Heating Refrigeration and Air-conditioning Engineers (ASHRAE). BACnet stands for **B**uilding **A**utomation and **C**ontrol **net**work. BACnet is a true, non-proprietary open protocol communication standard conceived by a consortium of building management, system users and manufacturers. It identifies all the rules for system components to share data with each other, how this is done, the communications media that can be used and how information from components is to be interpreted.

BACnet provides the method by which computer-based controls systems equipment from different manufacturers can work together, or "*interoperate*." It is designed to handle many types of building controls including HVAC, lighting, fire and other systems. The popularity of the concept is demonstrated by many projects that have incorporated multiple vendors' components using BACnet to communicate with each other.

LAN Technology

To meet the requirements of interoperability the BACnet committee incorporated several LANs into the BACnet standard. They chose existing LAN technologies whenever possible. In cases where LANs fitting the established criteria could not be found, the committee developed its own LANs. These LANs, listed in Table 1, cover a wide range from low cost to high performance.

BACnet LAN:	Standard:	Data rate:	Packet size:	Cost:
Ethernet	ISO/IEC 8802-3	10 to 100 Mbps	1515 bytes	High
ARCNET	ATA/ANSI 878.1	0.156 to 10 Mbps	501	Medium
MS/TP	ANSI/ASHRAE 135-1995	9.6 to 78.4 Kbps	501	Low
LonTalk	n/a	4.8 to 1250 Kbps	228	Varied

Table 1: BACnet LANs

Ethernet is a high-speed LAN that has been widely used for many years. By virtue of its popularity the expense of its interface has been dropping, though it still remains high compared to many other LANs. It offers a number of media alternatives such as twisted-pair, coax and fiber-optic cabling. Off-the-shelf interfaces for personal computer workstations are readily available and inexpensive, though the need for hubs and repeaters can increase the cost of the system.

ARCNET is popular with the process control industry. It is a lower-cost LAN than Ethernet, but requires dedicated communication integrated circuits (ICs) which keep its cost higher than some BACnet LANs. (It was on the basis of cost that ARCNET was chosen over otherwise competitive international LANs.) The ARCNET specification defines suitable media as including, but not limited to, coaxial, twisted pair (shielded and unshielded) and fiber optic cables.

The MS/TP (Master-Slave/Token-Passing) LAN was designed to make it possible for BAS manufacturers to build BACnet devices with the low cost necessary for BACnet's success in competing with proprietary LANs. By virtue of its simple interface and its communication rates MS/TP can be implemented on many standard microcontrollers without the added cost of dedicated communications ICs. The MS/TP LAN uses EIA-485 signaling over twisted-pair wiring.

MS/TP devices come in two varieties: Slaves and Masters. Slave devices are especially suited for the lowest-cost implementations but they lack the capability to initiate requests; they can only reply to messages from other devices. Master devices are able to initiate requests, but they must also be able to negotiate for a time slot in which to make their requests. This adds some processing and memory requirements to the Master device which can result in higher cost than the Slave.

0-7803-5446-X/99 $10.00 © 1999 IEEE

LonTalk was originally developed as a proprietary LAN; LonTalk devices used a special communications device which incorporated three microprocessors to handle the overhead. Recently it has been released as an open protocol. Although LonTalk was developed for the LonWorks protocol it provides a means for what it terms "foreign frame transmission." The BACnet standard makes use of this capability of LonTalk for transporting its "foreign" frames.

INTERNETWORKING

It is frequently necessary to have multiple networks in a single BAS installation. There may be too many devices to be connected to a single LAN, or the requirements of the installation might dictate the use of different types of LANs for different functions. When two or more networks are set up to communicate with each other the result is called an "internetwork." Internetworks may be comprised of similar networks linked together or, as in BACnet, they may contain different networks with different characteristics.

Internetworking in a building automation system allows the control engineer to keep the system's cost down. This concept of mixing network varieties is not new. Many existing proprietary building automation systems are in fact internetworks comprising different types of LANs. BACnet, though, provides the control engineer with the flexibility of selecting the types of LANs to be used in a particular BAS for the lowest system cost.

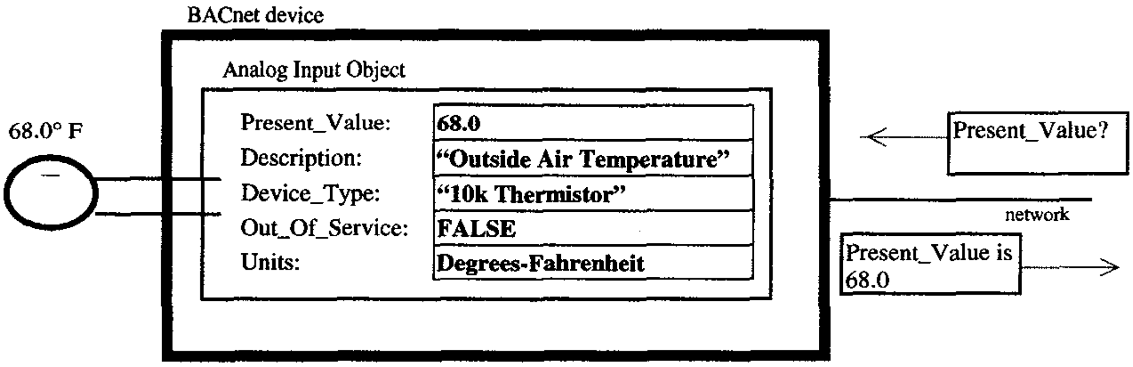
Objects

BACnet departs from traditional HVAC industry conventions with its object-oriented nomenclature. The industry has long used the general-purpose term "points", which could refer to sensor inputs, control outputs or control values, with different characteristics according to manufacturer. BACnet instead defines a standard set of *"Objects"*, each of which has a standard set of *"Properties"* which describe the Object and its current status to other devices on the BACnet internetwork, and through which the Object may be controlled by other BACnet devices. This is unlike a number of current systems wherein the actual description of a sensor may be distributed around the network.

One of the standard BACnet objects is the Analog Input Object, which represents an analog sensor input such as a thermistor. Figure 1 shows a diagram of just such an Analog Input Object as it might be seen over the network through five of its properties. Some of the properties, such as Description, Device_Type and Units, are set during installation. Others, including Present_Value and Out_Of_Service, provide status about the sensor input represented by the Analog Input Object. Yet others (not shown here; an Analog Input Object can have up to 25 Properties) may be set by the device's manufacturer. All may be read; in this example a query about the Present_Value Property of this Analog Object would get the reply "68.0".

BACnet defines 18 standard types of Objects, listed in Table 2. The list is intended to be comprehensive; each element of a complete building control system is represented by one or more of the Objects, whether Analog Input for a sensor, a Schedule for scheduling, or Notification Class for distributing alarms.

The choice of which Objects are present in a BACnet device is determined by the device's function and capabilities; the BACnet standard does not require all Objects in all BACnet devices. A device which controls a VAV box is likely to have several Analog Input and Analog Output Objects; a Windows™ workstation which has neither sensor inputs nor control outputs will not.



Present_Value:	**68.0**
Description:	**"Outside Air Temperature"**
Device_Type:	**"10k Thermistor"**
Out_Of_Service:	**FALSE**
Units:	**Degrees-Fahrenheit**

Figure 1. An Analog Input Object

Every BACnet device is required to have a Device Object, the Properties of which fully describe the BACnet device to the network. The Object_List Property of the Device Object, for example, provides a list of every Object contained within the BACnet device. The Vendor_Name, Vendor_Identifier and Model_Name Properties give, of course, the manufacturer and model of the device. BACnet allows manufacturers to provide proprietary Objects, though these proprietary Objects will not be necessarily accessible or understood by equipment from other manufacturers.

Table 2. Standard BACnet Objects:

Analog Input	Sensor input
Analog Output	Control output
Analog Value	Setpoint or other analog control system parameter.
Binary Input	Switch input
Binary Output	Relay output
Binary Value	Binary (digital) control system parameter
Calendar	Defines a list of dates, such as holidays or special events, for scheduling.
Command	Writes multiple values to multiple objects in multiple devices to accomplish a specific purpose, such as day-mode to night-mode, or to emergency mode.
Device	Properties tell what objects and services the device supports, and other device-specific information such as vendor, firmware revision, etc.
Event Enrollment	Describes an event which might be an error condition (e.g., "Input out of range") or an alarm that other devices might need to know about. It can directly tell one device, or use a Notification Class object to tell multiple devices.
File	Allows read and write access to data files supported by the device.
Group	Provides in a read single operation access to multiple properties of multiple objects.
Loop	Provides a standardized access to a "control loop."
Multi-state Input	Represents the status of a multiple-state process, such as a refrigerator's On, Off, Defrost cycle.
Multi-state Output	Represents the desired state of a multiple-state process (such as It's Time to Cool, It's Cold Enough. And It's Time to Defrost).
Notification Class	Contains a list of devices to be informed if an Event Enrollment object determines that a warning or alarm message needs to be sent.
Program	Allows a program running in the device to be started, stopped, loaded and unloaded, and reports the present status if the program.
Schedule	Defines a weekly schedule of operations (performed by writing to specified list of objects) with exceptions such as holidays. Can use a Calendar object for the exceptions.

Properties

The BACnet standard identifies 123 different Properties of Objects. A different subset of these Properties is specified for each type of Object. The BACnet specification requires for each Object that certain Properties must be present; other specified Properties are optional. In either case, the implemented Properties have specific behaviors defined by the specification; particularly those involved in alarm or event notifications, and those which have an effect upon control values or states.

A few of the standard Properties are required by the BACnet specification to be writable; others may be writable at the manufacturer's discretion. All may be read over the network.

BACnet does allow the addition of proprietary Properties by vendors, but as with proprietary Objects the proprietary Properties may not be understood or accessible by equipment from other manufacturers.

The Analog Input Object is representative of the Objects involved directly with control elements and many of its Properties reflect this. Table 3 lists the defined Properties of the Analog Input Object, along with typical or example values for each property. For example, the Status_Flags, Event_State, Reliability, Out_Of_Service, Min_Pres_Value, Max_Pres_Value, Notification_Class, High_Limit, Low_Limit, Limit_Enable, Event_Enable, Acked_Transitions and Notify_Type Properties all deal with the detection of unusual and possibly dangerous conditions at the sensor and generating the appropriate notifications or alarms in response.

Table 3. Properties of the Analog Input Object:

Object_Identifier	Required	**Analog Input #1**
Object_Name	Required	**"AI 01"**
Object_Type	Required	**Analog Input**
Present_Value	Required	**68.0**
Description	Optional	**"Outside Air Temperature"**
Device_Type	Optional	**"10k Thermistor"**
Status_Flags	Required	**In_Alarm, Fault, Overridden, Out_Of_Service** flags
Event_State	Required	**Normal** (plus various problem-reporting states)
Reliability	Optional	**No_Fault_Detected** (plus various fault conditions)
Out_Of_Service	Required	**False**
Update_Interval	Optional	**1.00** (seconds)
Units	Required	**Degrees-Fahrenheit**
Min_Pres_Value	Optional	**-100.0**, minimum reliably read value
Max_Pres_Value	Optional	**+300.0**, maximum reliably read value
Resolution	Optional	**0.1**
COV_Increment	Optional	Notify if Present_Value changes by increment: **0.5**
Time_Delay	Optional	Seconds to wait before detecting out-of-range: **5**
Notification_Class	Optional	Send COV notification to Notification Class Object: **2**
High_Limit	Optional	**+215.0**, Upper normal range
Low_Limit	Optional	**-45.0**, Lower normal range
Deadband	Optional	**0.1**
Limit_Enable	Optional	Enable **High-limit-reporting, Low-limit-reporting**.
Event_Enable	Optional	Enable **To_Offnormal, To_Fault, To_Normal** change reporting.
Acked_Transitions	Optional	Flags indicating received acknowledgments for above changes.
Notify_Type	Optional	**Events** or Alarms

The first three Properties listed, Object_Identifier, Object_Name and Object_Type are required to be present in every Object in a BACnet device.

The Object_Identifier is a 32-bit code which identifies the type of Object (also identified by the Object_Type Property) and its "*Instance*" number which together uniquely identify the Object within its BACnet device; theoretically a BACnet device could have over four million Objects of a particular type. The Object_Name is a text string which has a unique capability in that BACnet devices may broadcast queries for devices which contain Objects with a specific Object_Name. This can greatly simplify installation.

BACnet requires one *Device Object* to be present in every BACnet device. The Device Object makes information about the device and its capabilities available to other devices on the networks. Before one BACnet device starts controls-related communications with another, it needs to obtain some of the information presented by the other device's Device Object. Although the list of Properties is imposing, most are fixed by the manufacturer and read only by other BACnet devices.

Unlike other Objects, the Device Object's Instance number is required to be unique across the entire BACnet internetwork as this number is used to identify the BACnet devices, and may be used as a convenient reference to the BACnet device from other devices during installation.

Services

Services are the means by which one BACnet device acquires information from another device, commands another device to perform some actions, or announces to one or more devices that some event has taken place. Each *service request* issued and *service acknowledgment* (reply) returned becomes a message packet transferred over the network from the sending to the receiving device.

Service requests are issued, and upon receipt processed, by an "*application program*" running on the BACnet device. The application program is the actual software which performs the operations of the device. In the case of an operator workstation the software might be maintaining a display of several sensor inputs for the operator; it could periodically issue service requests to the appropriate Objects in the target devices to obtain the latest value of the inputs. In the monitored device the service request would be processed in its application program and the reply containing the requested data returned. This is diagrammed in Figure 2.

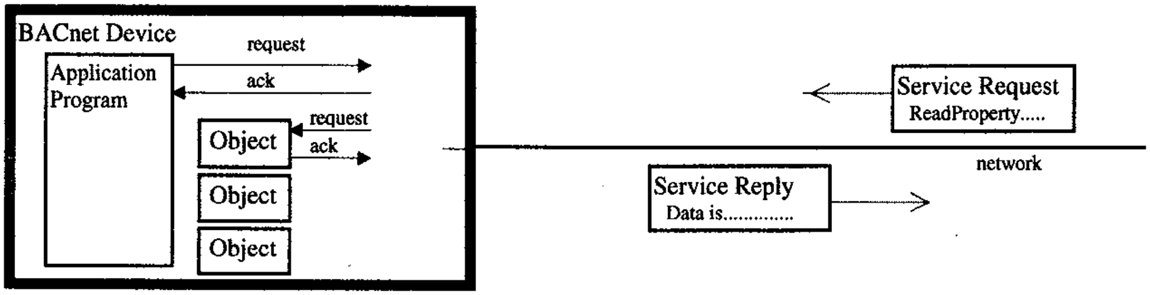


Figure 2. Service Requests and Replies

BACnet defines 32 Services and classifies them into five categories as shown in Table 4. These Service categories are the Alarm and Event, File Access, Object Access, Remote Device Management and Virtual Terminal Services. For each of the "Confirmed" (a reply, typically with data, is expected) services, labeled "C" in the tables following, the BACnet device may have the capability to initiate the Service request, or the ability to process and respond to a received request of that type, or both. For each of the "Unconfirmed" (no reply is expected) services, labeled "U" in the tables following, the BACnet device may have the capability to initiate the Service request, or the ability to process a received request of that type, or both.

File Access Services in BACnet are used to read and manipulate files in BACnet devices. In BACnet, files represent groups databytes of arbitrary length and meaning and do not necessarily relate to any kind of mass storage device. Every BACnet-accessible file has a File Object associated with it.

The Object Access Services provide the means to read, modify and write Properties, and to add and delete Objects. Multiple Services are provided for reading and writing properties; the purpose of the more complex Services (ReadPropertyMultiple and WritePropertyMultiple) is to combine as many reads of or writes to Properties of Objects within a BACnet device into a single message, thus reducing network overhead. The ReadPropertyConditional Service goes a step further; the device processing the request tests each referenced Property according to criteria included in the request and returns the value only if the criteria are met.

BACnet devices are not required to implement every single Service. Just one Service, ReadProperty, is required to be processed by all BACnet devices. Depending upon the function and complexity of the device, additional Services may be initiated or executed.

The Alarm and Event Services deal with changes in conditions seen by a BACnet device. This includes alarms; changes which might indicate problems or error conditions, such as a sensor reading out of normal range or, for that matter, returning to normal operation ("Events"); or a change in a reading (of some increment since the previous report), termed "Change Of Value" or COV.

The Remote Device Management Services provide a number of disparate functions, including operator control, specialized message transfer, and addressing/auto-configuring functions.

The Who-Is and I-Am Services are used to obtain the network addresses of BACnet devices on a BACnet internetwork; they can make life much easier for the installer by reducing or eliminating the need to program other devices' internetwork addresses into each BACnet device. Instead, a BACnet device which needs to know the address of one or more devices can broadcast a Who-Is Service request (message) on the internetwork specifying a Device Object Instance Number or a range of Instance Numbers. The responses do not come back as a reply. Instead the devices which have the specified Device Objects broadcast an I-Am Service request either on the local network, on a remote network, or on the entire internetwork, so that the requesting device will see the response which carries with it the address information of the responder. This allows other devices which may need to know about the responders to acquire the address information without creating more network traffic.

Table 4. BACnet Services

Alarm and Event Services

AcknowledgeAlarm	C	Used to tell sender of alarm that a human has seen the alarm.
ConfirmedCOVNotification	C	Tells subscribing devices of the COV that occurred in a property.
ConfirmedEventNotification	C	Used to tell sender of a possible error condition.
GetAlarmSummary	C	Requests from a device a list of "active alarms," if any.
GenEnrollmentSummary	C	Requests a list of "event" (possible error) generating objects.
SubscribeCOV	C	Sent by a device to request that it be told of COVs in an object.
UnconfirmedCOVNotification	U	Tells subscribing devices that a change has occurred to one or more properties of a particular object.

File Access Services:

AtomicReadFile	C	Requests part or all of a File object's file.
AtomicWriteFile	C	Writes to part or all of a File object's file.

Object Access Services:

AddListElement	C	Adds one or more items to a property that is a list.
RemoveListElement	C	Removes one or more items from a property that is a list.
CreateObject	C	Used to create a new instance of an object in the serving device.
DeleteObject	C	Used to delete a particular object in the serving device.
ReadProperty	C	Returns a value of one property of one object.
ReadPropertyConditional	C	Returns the values of multiple properties in multiple objects.
ReadPropertyMultiple	C	Returns the values of multiple properties of multiple objects.
WriteProperty	C	Writes a value to one property of one object
WritePropertyMultiple	C	Writes values to multiple properties of multiple objects.

Remote Device Management Services:

DeviceCommunicationControl	C	Tells a device to stop (and start!) accepting network messages.
ConfirmedPrivateTransfer	C	Sends a vendor-proprietary message to a device.
UnconfirmedPrivateTransfer	U	Sends a vendor-proprietary message to one or more devices.
ReinitializeDevice	C	Orders the receiving device to cold- or warm-boot itself.
ConfirmedTextMessage	C	Conveys a text message to another device.
UnconfirmedTextMessage	U	Sends a text message to one or more devices.
TimeSynchronization	U	Sends the current time to one or more devices.
Who-Has	U	Asks which BACnet devices contain a particular Object.
I-Have	U	Affirmative response to Who-Has, broadcast.
Who-Is	U	Asks about the presence of specified BACnet devices.
I-Am	U	Affirmative response to Who-Is, broadcast.

Conclusion

BACnet spent nine years under development by a committee drawn from manufacturers, universities, government agencies and consulting firms in an effort to produce a truly open protocol whereby equipment from different manufacturers can interoperate in a complete, integrated building automation control system. The result is a standard that defines all the elements of communication between devices, from the abstract language of Objects and Services right down to the physical LANs. With its adoption as an ANSI standard and the interest shown world-wide, it is safe to say that BACnet points the way to the future of communication within building automation controls.