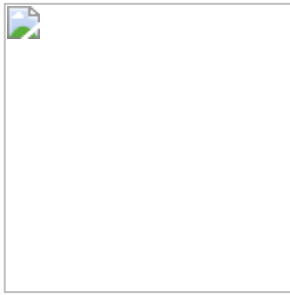


The Language of BACnet-Objects, Properties and Services

By Bill Swan, Alerton Technologies, Inc.



BACnet™ (Building Automation and Control Network) is the open data communications protocol which has been developed over the past nine years by ASHRAE. In December of 1995, BACnet was adopted by ANSI and is now an American National Standard (ANSI/ASHRAE 135-1995) and is under review by the International Standards Organization (ISO). BACnet was created to end the frustration caused by the inability of proprietary control systems to communicate with other systems within a building or other manufacturers' systems.

BACnet provides the method by which computer-based control equipment from different manufacturers can work together, or "*interoperate*." It allows you to expand, mix and match equipment to better fit any size of building's needs for the present and future. BACnet is designed to handle many types of building controls, including HVAC, lighting, security, fire, access control, maintenance, waste management and so forth.

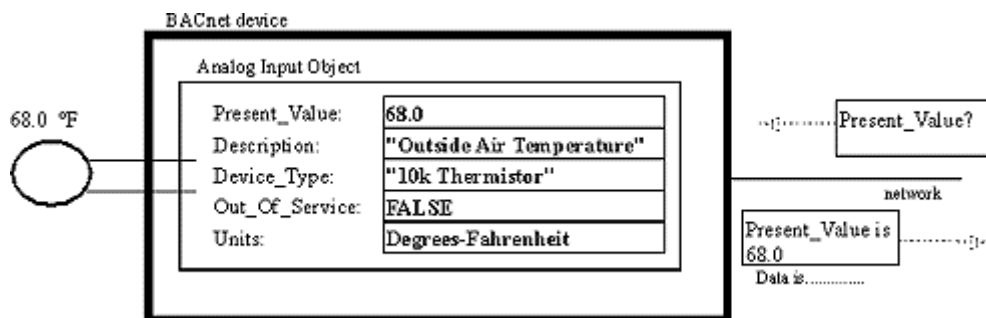
BACnet provides a sophisticated model for describing building automation systems of all types. This model is based on the ideas that for systems to be truly interoperable, there must be agreement about various aspects of the overall operation and the individual systems themselves. For the equipment to work together, components must be able to exchange and understand BACnet messages. BACnet specifies four types of local-area networks and a serial (EIA-232) interface along with networking protocols to get the messages from one device to another. The content of the messages, the BACnet language, is the focus of the major portion of the BACnet standard.

Objects

BACnet departs from traditional 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*"; that describe the Object and its current status to other devices on the BACnet internetwork. It is through these properties that the Object may be controlled by other BACnet devices.

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 (an Analog Input Object can have up to 25 Properties) may be set by the equipment manufacturer. All may be read; in this example, a query about the Present_Value Property of this Analog Input Object would get the reply "68.0".

Figure 1. An Analog Input Object.



BACnet defines 18 standard types of Objects, listed in Table 1. 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 that controls a VAV box is likely to have several Analog Input and Analog Output Objects while a Windows® workstation that has neither sensor inputs nor control outputs will not.

Every BACnet device must 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 provide the manufacturer name and model of the device.

In addition, BACnet allows manufacturers to provide proprietary Objects, which will not necessarily be accessible or understood by equipment from other manufacturers. However, they will not interfere with standard BACnet objects.

Table 1. Standard BACnet Objects.

OBJECT	EXAMPLE OF USE
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 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 that might be an error condition (e.g., "Input out of range") or an alarm that other devices 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 access to multiple properties of multiple objects in a read single operation.
Loop	Provides standardized access to a "control loop."
Multi-state Input	Represents the status of a multiple-state process, such as a refrigerator's On, Off, and Defrost cycles.
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	Contains a list of devices to be informed if an Event Enrollment object determines that a

Class	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 of 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 that certain Properties must be present for each Object; 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 that have an effect upon control values or states.

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

BACnet does allow vendors to add proprietary Properties 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 2 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 detecting of unusual and possibly dangerous conditions at the sensor and generating the appropriate notifications or alarms in response.

Table 2. Properties of the Analog Input Object.

PROPERTY	BACnet	EXAMPLE
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-must be present in every Object in a BACnet device.

The Object_Identifier is a 32-bit code that 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. BACnet devices may broadcast queries for devices that contain Objects with a specific Object_Name. This can greatly simplify project setup.

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. Table 3 gives an example of a Device Object for a BACnet Dual-Duct VAV controller. Although the list of Properties is imposing, most are fixed by the manufacturer and are read only by other BACnet devices.

Unlike other Objects, the Device Object's Instance number must be unique across the entire BACnet internetwork because it is used to uniquely identify the BACnet devices. It may be used to conveniently identify the BACnet device from other devices during installation.

Table 3. Properties of the Device Object.

PROPERTY	BACnet	EXAMPLE
Object_Identifier	Required	Device #1076
Object_Name	Required	"Office 36 DD Control"
Object_Type	Required	Device
System_Status	Required	Operational (plus others)
Vendor_Name	Required	"Alerton Technologies, Inc."
Vendor_Identifier	Required	Alerton
Model_Name	Required	"VAV-DD Controller"
Firmware_Revision	Required	"1.0"
Application_Software_Version	Required	"Dual-Duct DDC"
Location	Optional	"Office 36, Floor 3"
Description	Optional	"(on network 5)"
Protocol_Version	Required	1 (BACnet protocol version)
Protocol_Conformance_Class	Required	2
Protocol_Services_Supported	Required	readProperty, writeProperty, atomicWriteFile,...
Protocol_Object_Types_Supported	Required	Analog Input, Analog Output,...

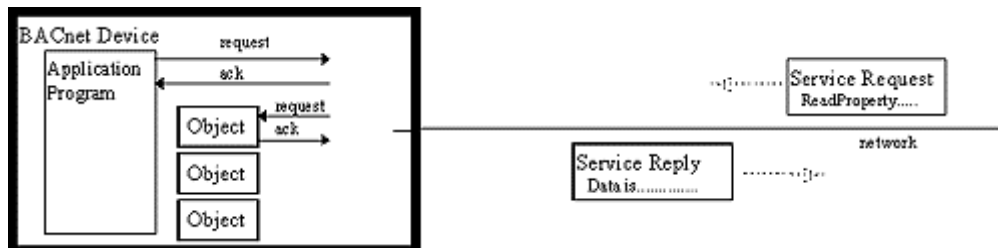
Object_List	Required	Analog Input #1, Analog Input #2, ...
Max_APDU_Length_Supported	Required	50 (bytes or characters)
Segmentation_Supported	Required	No
VT_Classes_Supported	Optional	n/a
Active_VT_Sessions	Optional	n/a
Local_Time	Optional	12:30:15.22
Local_Date	Optional	Tuesday, March 12, 1996
UTC_Offset	Optional	+480 (minutes from GMT/UTM)
Daylight_Savings_Status	Optional	False (not in effect)
APDU_Segment_Timeout	Optional	n/a
APDU_Timeout	Required	3000 milliseconds
Number_Of_APDU_Retries	Required	0
List_Of_Session_Keys	Optional	n/a
Time_Synchronization_Recipients	Optional	n/a
Max_Master	Optional	n/a
Max_Info_Frames	Optional	n/a
Device_Address_Binding	Required	None

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.

An "*application program*" running on the BACnet device issues service requests and processes them upon receipt. The application program is the actual software that performs the operations of the device. In the case of an operator workstation, the software might maintain a display of several sensor inputs for the operator and would 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 depicted in Figure 2.

Figure 2. Service Requests and Replies.



BACnet defines 32 Services and classifies them into five categories. These Service categories are the Alarm and Event, File Access, Object Access, Remote Device Management and Virtual Terminal Services. For each of the "Confirmed" services (a reply, typically with data, is expected), labeled "C" in the tables following, the BACnet device may have the ability 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.

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 including alarms. Alarms and events 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.

COV reporting is a useful alternative to repeatedly polling an Object for some monitored value. A device containing a monitored Object could be subject to network traffic congestion if many other devices were monitoring it; COV reporting allows it to send out notices when a change occurs.

Alarm, Event and COV notifications are generated by the application program monitoring Objects directly related to control operations: specifically, the various Input, Output and Value Objects, as well as the Loop Object.

Table 4. Alarm and Event Services.

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

The word "Atomic" in the service name merely means that only one read or write operation can take place at a time. Other attempts to access the file during an access will either fail or be held off.

At present there are no standard BACnet-defined files; by definition, all file accesses are vendor proprietary though BACnet has reserved the EVENTLOG and VALUELOG file types for file structures to be defined at a later date.

Table 5. File Access Services.

SERVICE	BACnet	DESCRIPTION
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.

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.

Although the CreateObject and DeleteObject Services are defined, they will apply to a limited set of Objects. Objects attached to some physical device, for instance, are not likely to be either creatable or deletable. The Group and Event Enrollment Objects, and possibly the File Object, are likely to be the usual subjects of the CreateObject and DeleteObject Services.

Table 6. Object Access Services.

SERVICE	BACnet	DESCRIPTION
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.

The Remote Device Management Services, listed in Table 7, provide a number of disparate functions, including operator control, specialized message transfer, and addressing/auto-configuring functions.

The DeviceCommunicationControl and ReinitializeDevice Services are intended to provide a human operator with diagnostic tools which may be invoked remotely. With them a BACnet device can be ordered to ignore all BACnet messages (except for the DeviceCommunicationControl and ReinitializeDevice Services), or either warm-started or cold-started.

The ConfirmedPrivateTransfer and UnconfirmedPrivateTransfer Services are used to convey messages outside the BACnet standard, effectively invoking non-standard Services. These Services bear both the Vendor ID Code and the Service Code in standard BACnet form; the rest of the content is entirely up to the vendor and not likely to be interpretable by other manufacturers' devices.

The ConfirmedTextMessage and UnconfirmedTextMessage Services transport text messages to other devices, such as printers.

The TimeSynchronization Service is transmitted by designated devices which have clocks to other devices, or broadcast, so that the devices may be synchronized.

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 that 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 that 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 that may need to know about the responders to acquire the address information without creating more network traffic.

The Who-Has and I-Have Services are similar to the Who-Is and I-Am, but the Who-Has adds either an Object Identifier (Object type plus Instance number, unique within each device) or an Object Name (a Property of every Object, unique within each device). Receiving devices which contain an Object matching the request broadcast the I-Have Service request.

Table 7. Remote Device Management Services.

SERVICE	BACnet	DESCRIPTION
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.

The Virtual Terminal (VT) Services can be used by an operator to establish a bi-directional text-based connection with an application program executing in a remote device. In effect, for the duration of a VT session established with the remote device, the operator's device looks like a terminal connected to the remote application program.

Table 8. Virtual Terminal Services.

SERVICE	BACnet	DESCRIPTION
VT-Open	C	Establishes a virtual terminal session with a remote BACnet device.
VT-Close	C	Closes an established virtual terminal session.
VT-Data	C	Sends text from one device to the other participating in a session.

[Note: Due to changes in the BACnet standard years after this article was published, the following section is out of date and no longer applies. Please see Annexes A, B, K and L in BACnet-2004. --Bill Swan]

Conformance Classes, Function Groups and the PICS:

Evaluating the capabilities of a BACnet device is potentially a formidable task, given the great choice of Objects, Properties and Services which can be implemented, as well as the fact that it is not necessary for every BACnet device to have a full BACnet implementation in order to carry out its task. ASHRAE's BACnet Committee recognized this problem and responded with aids to evaluation in the form of "Conformance Classes," "Function Groups" and the "Protocol Implementation Conformance Statement" (PICS).

The BACnet protocol defines six levels of Conformance Classes, each of which specifies the minimum subset of Services implemented on the device. The lowest level, Conformance Class 1, requires only that the BACnet device contain a Device Object and that it be able to execute (respond to) a ReadProperty Service request. Each successive Conformance Class level adds Service Requests that must be executable by the device, as well as the Service Requests it must be able to initiate. Conformance Class 6 requires 21

types of Service Requests (of the 32 overall) to be implemented, of which 20 must be initiatable and 17 executable. Conformance Class thus provides a measure of the device's ability to communicate.

Function Groups specify a combination of Objects and Services necessary to carry out certain building automation functions. They are specified independently of Conformance Class, though the implementation of some of the Function Groups automatically confers some Conformance Class higher than 1. The Function Groups specified by BACnet are listed in Table 9, with brief descriptions.

Table 9. Function Groups.

FUNCTION GROUP	DESCRIPTION
Clock	Capabilities generally associated with having a clock.
Hand-held Workstation	Hand-held or portable workstation capabilities.
Personal Computer Workstation	Main operator workstation capabilities.
Event Initiation	Detect and generate alarms and events.
Event Response	Ability to respond to alarms and events.
COV Event Initiation	Ability to initiate change-of-value notifications.
COV Event Response	Subscribe to and receive change-of-value notifications.
Files	Capability to read, write, upload and download files.
Reinitialize	Ability to be reinitialized from remote device.
Virtual Operator	Capable of providing operator side of virtual terminal session.
Virtual Terminal	Capable of providing server side of virtual terminal session.
Device Communications	DeviceCommunicationControl Service request execution supported.
Time Master	Able to initiate TimeSynchronization Service request.

In order to best facilitate evaluations of BACnet devices, the BACnet specification also provides a standard Protocol Implementation Conformance Statement (PICS), created by the manufacturer, which provides in detail the options implemented in the particular device. It identifies the vendor, briefly describes the device, and details of the implementation. Included in the PICS are:

- the Conformance Class of the device,
- the supported Function Groups,
- standard and proprietary Services executed and initiated,
- standard and proprietary Objects with any optional, writable, creatable, or deleteable Properties

and a number of other aspects of the device pertaining to its implementation per the BACnet standard. The Standard provides a pro forma PICS to simplify the effort of specification.

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 by GSA and others, it is safe to say that BACnet points the way to the future of building automation controls.