FOUNDATIONS
A BACnet INTERNATIONAL PUBLICATION

This article was published in Foundations, A BACnet International Publication, August 2013. © BACnet International 2013 – further editorial use of articles in Foundations is encouraged. Please send a copy to the BACnet International office at info@bacnetinternational.org.

BACnet Developers Q & A

By Steve Karg, Member ASHRAE

Over the years while being involved in the BACnet committee and developing BACnet products, I have fielded questions about BACnet development. Some of those questions are answered by the official BACnet Testing Laboratories "Implementation Guidelines". However, some questions are beyond the general scope of that document.

Question: I have created a BACnet device. The device is discovered in the BACnet network. Since my device has no DIP switch or user interface, is it possible to set the Device object instance number from the BACnet network?

Answer: I have also created BACnet devices without DIP switches or a user interface, and have enabled all of their properties configurable from the BACnet network. To configure the Device object instance number, you could have a writable Object_Identifier property in the Device object, or you could model an Analog Value object with its Present_Value property as the Device object instance number. Using the Analog Value object may allow more generic workstation devices to write the value, while using the Device object is more compact and direct.

Question: In clause 21, FORMAL DESCRIPTION OF APPLICATION PROTOCOL DATA UNITS, of the BACnet standard, the productions are defined for the Application datatypes in clause 20.2.1.4, Application Tags. The definition for Unsigned is "[APPLICATION 2] INTEGER (0..MAX)" What is MAX?

Answer: MAX is a reserved word from the ASN.1 standard. The ASN.1 is defined in several documents. The first clue to MAX comes from INTERNATIONAL STANDARD 8824-1, ITU-T RECOMMENDATION X.680, INFORMATION TECHNOLOGY –ABSTRACT SYNTAX NOTATION ONE (ASN.1): SPECIFICATION OF BASIC NOTATION.

> 47.5 Size Constraint
> 47.5.1 The "SizeConstraint" notation shall be:
> SizeConstraint ::= SIZE Constraint
> 47.5.2 A "SizeConstraint" can only be applied to bit string types, octet string types, character string types, set-of types or sequence-of types.
> 47.5.3 The "Constraint" specifies the permitted integer values for the length of the specified values, and takes the form of any constraint which can be applied to the following parent type:

> INTEGER (0 .. MAX)
> The "Constraint" shall use the "SubtypeConstraint" alternative of "ConstraintSpec".

The second clue to MAX comes from INTERNATIONAL STANDARD 8825-3, ITU-T RECOMMENDATION X.692, INFORMATION TECHNOLOGY – ASN.1 ENCODING RULES – SPECIFICATION OF ENCODING CONTROL NOTATION (ECN):

> 16.2.11 "MIN" and "MAX" specify that there is no lower or upper bound respectively.
> "MIN" shall not be used in "Size". "Fixed" means a single value or a single size.
> "SignedNumber" is specified in ITU-T Rec. X.680 | ISO/IEC 8824-1, 18.1. It shall be non-negative when used in "Size".
> "ValueOrMin" and "ValueOrMax" specify lower and upper bounds respectively

In my experience, 32-bit is an adequate size for MAX, but some vendors are starting to support 64-bit. I suppose in a few more years it could be even be larger.

Question: I am trying to understand when "Context Specific" would be used over "Application." Do you have any suggestions? I see the "WhoIs" service uses it, but could the service also use "Application" in leu of?

Answer: In most cases, you are required to follow the BACnet standard in regards to "Context Specific" or "Application" tagged data. The ASN.1 shows "Context Specific" in brackets (e.g. [0]) next to data that has context tags.

The only time you have a choice in the matter is when you are creating proprietary objects or proprietary properties (i.e. vendor invented data), or using PrivateTransfer services (i.e. vendor invented service).

About the Author

Steve Karg is a Senior Engineer at WattStopper, in Birmingham, Alabama. He has been an active member of ASHRAE SSPC 135 (BACnet) since 2001, and convenes their Lighting Applications working group. He wrote the open source BACnet Protocol Stack hosted on SourceForge.net, and continues to help maintain the BACnet decoder in Wireshark.