

Source: Erik Ivanenko on BACnet-L (27-Jun-2006)

Question: *If a device supports alarm annunciation, AE-NI-B for example, will it also support alarm distribution?*

I have a need in which an alarm is to be announced at a remote location during the day, and a different location at night.

I cannot find a BIBB that would cover such behaviour. Where would this alarm routing/targeting be implemented: at the source of the alarm or would a B-OWS be responsible for doing this?

Answer: David Fisher

This question elicited a thread of responses that are reproduced here...

Brad Spencer observes:

The requirement of the presence of the notification class object for those devices supporting AE-NI-B would ensure your time based distribution requirement through the use of the recipient list property. The configuration of the recipient list is of course the right of the B-OWS.

Like many things in the standard there may be no explicit requirement that answers your question however recognizing common sense with the goal of interoperability in mind it would be safe to say the following.

"A device that supports AE-NI-B must support a notification class object along with device address binding in order to interoperate. In order to interoperate all recipient list parameters and combination of parameters must be writable and all combinations of binding that would allow all bindings must be supported."

If a device does not support the all forms of recipient list writability and is unable to provide device address binding in all forms then the device has not been implemented to interoperate and should not be considered a proprietary alarming solution.

Answer: David Fisher

In my opinion, the advice in the reply sent to you (from Brad Spencer) is wrong on all counts.

There is no litmus test for interoperation. The whole purpose of BACnet is, among other things, to define a collection of possible kinds of interoperations, and device designers may choose those interoperations that they want to provide.

For two BACnet devices to "interoperate" it is not, and was never promoted or intended to be, an "all possible combinations" thing. Let's use your use-case as an example.

You want the distribution of an alarm notification to reach different destination devices at different times. In order for that kind of use-case to be satisfied, there are several things that have to happen:

1. The server (the device that detects the alarm) must be able to initiate an EventNotification, AND that EventNotification must be able to be directed to a specific device or devices.

2. The destination device(s) (clients) must be able to receive an EventNotification.
3. It must be possible to configure the server device so that it knows to initiate an EventNotification once the alarm is detected.
4. Either of these two things must be true:
 - the server supports a mechanism through configuration that allows the selection of two destinations based on time, or
 - the server supports a single destination which is a BACnet client that knows how to redistribute the notification to different destinations based on time.
5. The standard provides ONE WAY that COULD BE used to represent such configuration information using a Notification class object. IF the NC object supports more than one recipient, and IF the NC object supports the flexibility of different times in the recipient, then this would work to represent in a BACnet-visible manner what the configuration was that had been made (through some means).
6. IF such an NC object is supported in a server, and IF the NC object supports multiple recipients, and IF the configuration of those parameters is writable through those properties of the NC object, then it would ALSO be possible to configure such a device for the purposes you've outlined.

NONE of these behaviors is required for a device that initiates EventNotifications, nor are they required for support of AE-N-I-B.

Most importantly, the BIBB describes the interoperation of sending and receiving of an alarm notification, period. It doesn't say or need to say, anything about how it's configured, and the depth of the implementation with regard to alarm distribution.

Don't get me wrong, those are useful things to be able to describe using BIBBs, but the current standard does not define BIBBs for those additional interoperations:

- destination specification
- NC object writability or scope of implementation

etc.

To say that a device can't interoperate without ALL of these criteria and more, is just wrong. This misrepresents what interoperability as conceived of and defined in BACnet means. It simply means that we define each "interoperation" and what each side needs to be able to do to use that interoperation.

The previous commentary (from Brad) presumes, incorrectly, that the standard defines a package of detailed criteria for multiple interoperations under the AE-N-I-B BIBB and that isn't what it does, nor is it required for the basic interoperation to work.

It is possible, as in this example, to describe a use-case that can be solved using multiple approaches, some of which may depend on multiple interoperations. But it's just wrong to say that the only useful definition for interoperability is to support every combination of everything for every use-case.

Speaking directly to your point:

- Yes, you want the server device to support AE-N-I-B so that you are assured that it can initiate an EventNotification
- If you want to have the server itself determine the destination routing, AND you want to assure that you can change this through BACnet simple services like WriteProperty, then you'll need to add criteria that says:
 - The server device must support NC objects that have multiple possible recipients
 - The server device's NC object must support writing to the Recipient_List property
 - The server device's NC object must implement the time-based features for destination qualification for each recipient

Alternatively you may want to consider having a single destination and use an alarm server device there which can do redistribution. The downside to this approach is that since BACnet does not define a formal mechanism for an alarm distribution server, it would invariably be a proprietary alarm server. The upside is that all devices could have simplified configuration since only one alarm destination is required for all alarm initiating devices. This also allows a simpler (less expensive) class of alarm initiating device to be used, and less distributed configuration and maintenance.

Either way, your system will most definitely be "interoperable" in that it will fulfill the mission of delivering an alarm to a destination client (the interoperation).

Chris Gurtler observes:

1. *It seems that your interpretation of a list means an implementor can implement it as an item, or just leave it empty. Not exactly what I call a list, surely a list should be capable of containing more than 1 item, perhaps 2 at least. There should be a minimum quantity greater than 1.*
2. *A destination under your interpretation does not need to support ValidDays, FromTime, ToTime etc. I'm not sure how you came to this conclusion, but based on your email, it looks like a device can ignore certain parameters of the destination. A device should support all parameters of the destination, not just some of them.*

Answer: David Fisher

1. Show me where the standard says that the recipient list has to contain anything let alone "more than one" entry. The list exists so that devices can provide a BACnet-visible picture of what their configuration is.

It's true that some devices may also allow configuration through this list, but that isn't a requirement of the standard. If it was, the standard would make that a required writable property, which it doesn't. If there was a requirement that the list have a minimum size then the standard would say that also. Let's not confuse what we think it should have said with what it actually says.

In terms of advising an end-user about expectations, we can't say that just because a device can initiate an event notification, that it follows as a requirement that there is support for these other things. We can espouse the view that those are desirable behaviors which ought to have been (or should be in the future) made into requirements (a view that I'm OK with)

but that isn't the requirement today. It also isn't a requirement to achieve the interoperability of alarm sending and receiving which is what the BIBB is about.

2. The standard, in describing the NC object's recipients, does not say that the fields of those recipients are required to be writable, nor does it say that all combinations are required to be implemented if it is writable. Again the concept was to be able to represent existing proprietary concepts of recipients and conditional recipients in BACnet-visible terms. If a device just can send an alarm, but not conditionally based on time or days of week, then it can represent those conditionals as all-inclusive values.

Again, I think it would be great if NC objects were required, and required to be writable in all of those respects, but that wasn't the consensus when the standard was made, so that's why the language is non-specific in this regard.

I couldn't say what the consensus would or wouldn't be if these issues were put to the test again today. For that to happen, someone would have to draft a proposal for making the multiple changes to the standard that would result in adding these kinds of requirements. That would also affect the test standard.

Brad Spencer observes:

(quoting from the BACnet standard)

L.3 BACnet Advanced Application Controller (B-AAC)

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*

It is a requirement of the BACnet standard that devices generate event notifications and allow for alarm parameters to be adjusted. The objects section is trumped by the all powerful profile section and not the other way around. For B-BCs it is a requirement that the device be designed so that it is general purpose so I am having difficulty understanding how you could argue your case without ignoring profiles. Because the "W" does not appear in the objects section of the standard does not mean it is not a requirement of the standard. Conformance to an interoperable profile demands interoperability.

In my opinion your arguments consistently paint a picture of devices that do not conform to profiles. If you want to find loopholes in the standard and you want to make things like recipient list proprietary because there is no "W" next to the property in the standard then you need face up to the fact that you have not developed a B-BC or a B-AAC and you may NOT market your device otherwise. Profile BACnet devices do things the native and interoperable BACnet way.

Answer: David Fisher

(speaking to Brad) In this reply, you've used the term "you" and "your" nine times, it appears trying to imply that I am making BACnet devices with those characteristics that you feel are not interoperable. Please don't do that anymore. The design choices that you or I might make aren't the issue.

I am reminding everyone about what the standard says. There are implications from the language in the standard, and some of them are subtle. I think it is helpful to those less familiar with the standard to see those implications or side effects. Some implementations of BACnet

take a more minimal view and some take a more maximal view of what they should offer in their products. I'm not making a judgment, just stating a fact.

Some buyers/users may feel that more is better.

(quoting Brad) "It is a requirement of the BACnet standard that devices generate event notifications and allow for alarm parameters to be adjusted."

No. There is no requirement that "devices" as in ALL devices be capable of generating event notifications. Only devices that claim to support in this case the BIBB AE-N-I-B are required to generate event notifications. If a device ALSO claims to support the B-AAC profile (for example) it must support the AE-N-I-B BIBB and by inference also the generation of those notifications. It's true that the B-AAC profile specifies that alarm parameters must be adjustable, but it doesn't say how that adjustment takes place and it doesn't say that it must be adjustable on-the-fly through BACnet services such as WriteProperty.

Those things (BACnet alarm delivery adjustment configuration through services like writing to properties) are good to have, but they don't affect the ability of a device to interoperably deliver an alarm notification.

(quoting Brad) "The objects section is trumped by the all powerful profile section and not the other way around."

There is no trumping involved. The profile simply states the goals of devices claiming to support the named profile. All that the profile does is put a name on a collection of BIBBs, and provide some vague guidance about the implementation of those BIBBs.

That's the reason, IMHO, that not everyone is enamored about profiles. They are vague and non-prescriptive and do not spell out in detail what expectations one should have about devices claiming to conform to them. At least BIBBs have some specificity.

Profiles aren't all-powerful, they are perhaps all-inclusive in the sense that they say what BIBBs must be provided. Regrettably in their current form profiles do not spell out the kind of detail necessary to really define all of the things an implementor needs, or for that matter a buyer should expect.

(quoting Brad) " For B-BCs it is a requirement that the device be designed so that it is general purpose"

What does "general purpose" mean and where is it defined what the criteria are against which a device's general purpose-ness is measured?

(quoting Brad) "so I am having difficulty understanding how you could argue your case without ignoring profiles."

I haven't ignored profiles at all. In the B-AAC example profile cited, there is no description that says anything about writing to properties of BACnet objects in order to make adjustments, or configuration, or determination of what recipient BACnet devices are. It just says that the B-AAC device must provide some way to achieve that. The BIBBs that are listed in L.7 do not include specificity for such configuration interoperations. The reason for that is that (notwithstanding the fact that Brad and I happen to both think this would be a good idea) there was no consensus on configuration as a requirement for interoperability (at least for this aspect of alarming) when the standard was made and throughout its subsequent numerous revisions.

(quoting Brad) "Because the "W" does not appear in the objects section of the standard does not mean it is not a requirement of the standard."

With all due respect, that's exactly what it means. Clause 12 of the standard in its introduction, defines this exactly (red emphasis added by me to highlight the relevant statements):

(quoting the BACnet Clause 12 preamble) "The summary includes the property identifier, the datatype of the property, and one of the following : O, R , W

*where O indicates that the property is optional,
R indicates that the property is required to be present and readable using BACnet services,
W indicates that the property is required to be present, readable, and writable using BACnet services.*

*When a property is designated as required or R, this shall mean that the property is required to be present in all BACnet standard objects of that type. When a property is designated as optional or O, this shall mean that the property is not required to be present in all standard BACnet objects of that type. The value of R or O properties may be examined through the use of one or more of the ReadProperty services defined in this standard. Such R or O properties may also be writable at the implementor's option unless specifically prohibited in the text describing that particular standard object's property. **When a property is designated as writable or W, this shall mean that the property is required to be present in all BACnet standard objects of that type and that the value of the property can be changed through the use of one or more of the WriteProperty services defined in this standard...**"*

In every context of the standard where standard objects have properties that are required to be writable, they are noted with W, or footnotes that call out circumstances when a normally not required to be writable property must become writable, e.g. Present_Value when Out_Of_Service is TRUE.

(quoting Brad) "Conformance to an interoperable profile demands interoperability. "

Being truthful demands making true statements. Of course. But being able to read properties of objects, doesn't demand being able to receive alarms because those are two different interoperations. A device that can read a property isn't "not interoperable" if it can't also receive and alarm. BACnet defines multiple interoperations, and BIBBs for each so that one can know what interoperations a device can and can't do. Profiles define particular collections of interoperations that may be supported, and some minimal information about the expected implementation of them.

But it is reading much more into the standard than is there to jump past that to some particular set of expectations that "seem reasonable" based on a sentence in a profile. I'd be the first to agree that a number of the profile's goal statements are vague, and it would be in the interest of everyone to make them more specific. So far no one has made a sweeping proposal to revise Annex L in that manner.

(quoting Brad) "In my opinion your arguments consistently paint a picture of devices that do not conform to profiles."

I'm just reporting that the standard, as written, doesn't specify the detail required to support the conclusions as you've presented them. Because Annex L is vague in some regards, various implementations have inconsistent interpretations of what "conformance" means. Historically

the interpretations have been mostly literal, in that if it doesn't say something explicitly, then its not a strict requirement. Again, that's what some implementations do.

(quoting Brad) "If you want to find loopholes in the standard"

For the record, I'm not personally attracted to "loopholes" nor do I view their exploitation as a best practice. However, the standard is what it says, not what it should have said. Many of the dozens of outstanding proposals before the SSPC are attempts to rule on issues of loopholes, generally for the better interest of end-users.

(quoting Brad) "and you want to make things like recipient list proprietary because there is no "W" next to the property in the standard ..."

Again, for the record, I don't want to make anything "proprietary." If a given implementation chooses to not implement writable NC object properties, that doesn't make them "proprietary" as this is allowed in the standard because the standard does not explicitly require it. If I have an Analog Input that has a fixed Description (for example in a box where AI-5 is ALWAYS the duct temperature sensor, it has a Description "Duct Temperature") that doesn't make the device or even that object "proprietary."

When there is a consensus in the development of a part of the standard that a given standard object property must always be writable, it is so indicated in the standard.

(quoting Brad) "...then you need face up to the fact that you have not developed a B-BC or a B-AAC and you may NOT market your device otherwise."

There is no directive in the profiles for B-BC or B-AAC that supports these assertions. If the SSPC reaches a consensus that required writability of specific properties is a good thing, then the appropriate language in the object and/or BIBB and/or profile can reflect that requirement. The absence of that language indicates a lack of consensus on that point. Since there is no such requirement, I don't see how you get to this conclusion.

(quoting Brad) "Profile BACnet devices do things the native and interoperable BACnet way"

Devices conforming to specific profiles, have nothing to do with "nativeness" or necessarily to their interoperability. A device conforming to B-ASC can't interoperate for the purposes of ReadPropertyMultiple with a B-BC that tries to send it a ReadPropertyMultiple. That doesn't invalidate either device's interoperable-ness. Both can conform strictly even to a lavish expectation of behavior for each profile and still not achieve that.

Native devices simply use BACnet as their means of communication without requiring a stand-alone gateway device. That has nothing to do with their interoperability or their conformance to a specific profile.

I think that the sentiment behind this statement is that BACnet devices should go to every reasonable length to maximize the scope of other devices that can use them. That's a fair goal, but often implementors run into a quandary where they must choose implementation limits even though the standard does not rule on what they might be.

(speaking to Chris's comments about lists) Coleman Brumley observes:

From the introduction section of Clause 12 in 135-2004:

*A "List of" datatype is a structured datatype consisting of a sequence of **zero or more** data elements, each having the same datatype. The length of each "List of" may be variable. Unless specified for a particular use, no maximum size should be assumed for any "List of" implementation. The notation "List of datatype" shall mean a sequence of **zero or more** data elements, each of which has the indicated type.*

*So, lists **in general** can be empty.*

*However, in the case of the recipient-list property, section 12.21.8 says that the list "shall convey a list of one or more recipient destinations". So, there is a minimum quantity of one – **for the recipient-list**. So, a recipient-list with one entry is as valid as a recipient-list with 4 entries.*

Also, Table 12-25 states that the ValidDays component of a BACnetDestination "may" be used, with emphasis on "may". My read on that is that valid days can be ignored. If you're implying that the ValidDays component can optionally be present in the encoding, that's not correct and certainly not what David is saying. But, based on table 12-25 the value of ValidDays can be ignored so a device may be ignoring the value.

(quoting Brad) If a device does not support the all forms of recipient list writability and is unable to provide device address binding in all forms then the device has not been implemented to interoperate and should ~~not~~ (SIC) be considered a proprietary alarming solution.

If my device supports write property and not Add/Remove list element, it's proprietary? If Device A and Device B can both support writing to a list via write-property, they're not interoperable? That's simply not true. It is true that if Device A supports only write property, yet device B only supports AddListElement et al, then they are not interoperable in that regard. That, in and of itself doesn't make either device lame. Nor does it make them proprietary. They still "speak BACnet".

IMO, raising the bar to require a device to support "all forms" like this is a shotgun approach to interoperability. And, before the emails start, I know that BTL requires this behavior. It doesn't mean I agree with it. I, personally, have never been a supporter of the "raise the bar" arguments. It may be a good idea in some cases. More often than not, it leads to the "what 135 says and what BTL says" types of scenarios. Now, it appears as though some think that if a device isn't implemented the BTL way, it's proprietary. Now, I have to ask, how is propagating that type of misinformation providing a service to anyone -- end-user or not?

Erik observes:

The exact description for ValidDays is: "The set of days of the week on which this destination may be used between From Time and To Time".

I read this as meaning that one cannot count on this feature being functional. Therefore, the target of the alarm must do the routing to the alarms device.

Since the target is a program handle, and alarm devices could be anything, alarm routing is clearly proprietary. It is clearly a configuration issue.

It is becoming apparent that my spec **shall** contain only the least common features.

From here on in I will read any feature that states “may” as being equivalent to “will be ignored during BTL certification”, or “is unlikely to be implemented by the vendor”.

Brad observes:

Except for some configuration requirements needing to be applied to the B-BC I think the arguments that were being applied to alarming are rare among manufacturers. As much as the BTL needs to be blamed for certain things they can not be blamed for alarming because BTL alarming is pretty interoperable and they have taken a very thoughtful do all approach to the full binding and alarming transaction. Alarming and binding is not as fault tolerant as it should be because more negative test should be applied to products however for the most part alarming is solid.

Erik observes:

1) Suppose a B devices point objects support intrinsic alarming, and claims AE-NI-B.

Suppose further, that this B device shall issue an alarm notification to program 17 on device 12.

Is the specification that the ‘B’ devices supports DM-DDB-A sufficient for me to know that the device will issue a “who-is” request to locate device 12?

Or can this ‘B’ device require that the location of device 12 be somehow hard-coded and still claim DM-DDB-A conformance? If so, device 12 should never be re-addressed! Note: this appears to indicate that for robust alarming, there ought to be an alarm device on each network segment! The alarm device will route the alarm (see 3).

2) Note that the B-AAC profile specifies AE-NI-B, but not DM-DDB-A, so the device address must be hard-coded. Could a device still issue a “who-is” request and not claim DM-DDB-A?

3) Also note that only the B-OWS claims alarm routing as a function and its profile lists AE-N-A. Does that mean all AE-N-A conformant devices do the alarm routing?

I believe that the BIBB statements, while important, should be extended to include specific behaviours. I do not want to work at BTL, or pretend that I am working at BTL, yet it appears that the responsibility of interpreting the standards document is a “local matter” that I must shoulder

Answer: David Fisher

(quoting Erik) Is the specification that the ‘B’ devices supports DM-DDB-A sufficient for me to know that the device will issue a “who-is” request to locate device 12?

Generally in those contexts where the device would act as an initiator who is trying to send to another device whose device instance is known, but not whose MAC address is known, the answer is YES.

But then you ask a trick question:

"Or can this ‘B’ device require that the location of device 12 be somehow hard-coded and still claim DM-DDB-A conformance?"

That’s a different issue. When the device is being configured to know the destination for an event notification, the standard allows this configuration to occur in various ways, as we’ve already covered in detail.

In those cases when the destination is specified using a NC object, then the standard allows the recipient to be expressed either as a (network number, MAC address) pair, or as a device instance. If the “device instance” form of destination is used then the device would also have to support DM-DDB-A in order to be able to do dynamic binding (discovery of the device instance-to-mac address relationship). If the device already knows the binding, then it wouldn’t need to use DM-DDB-A.

As you point out, the bad feature of using a fixed (net,mac) destination is one of maintenance. If that address ever changed you would have to hunt down all recipient programming and change it in every device, NC object etc. Obviously that would be a headache.

On the other hand, when you use the device instance form, the onus is on the initiator to discover the destination’s (net,mac) on the fly. Smarter devices can do this once and keep it around.

But the short answer to your implied question is that NO, just because you support DM-DDB-A it doesn’t mean you are required to be capable of using the device instance form of event notification recipient specification. Although in practice it probably is usually that way.

(quoting Erik) ... this appears to indicate that for robust alarming, there ought to be an alarm device on each network segment!

No. The DM-DDB-A discovery process will find the recipient you need on any network segment because Who-Is broadcasts are global as are I-Am replies.

(quoting Erik) ...2) Note that the B-AAC profile specifies AE-NI-B, but not DM-DDB-A, so the device address must be hard-coded.

You can be a B-AAC and still implement DM-DDB-A. It just means that devices that don’t support device instance recipients can still be B-AACs.

(quoting Erik) ... Could a device still issue a “who-is” request and not claim DM-DDB-A?

Yes

(quoting Erik) ...3) Also note that only the B-OWS claims alarm routing as a function and its profile lists AE-N-A. Does that mean all AE-N-A conformant devices do the alarm routing?

(also from later email: B-OWS profile says AE-N-A, and the text of operation says it is responsible for ‘adjustment of alarm routing’. I suspect the B-OWS would also be forced to route the alarm?

No. The “adjustment of alarm routing” referred to in B-OWS means that a B-OWS should be able to make changes to recipients and destination criteria in those devices that support that adjustment. There is no expectation that B-OWS will receive and resend alarms to other devices.

(quoting Erik) ...I believe that the BIBB statements, while important, should be extended to include specific behaviours.

There are some people who agree with you about that. Some proposals along these lines are in progress.

(quoting Erik) ...I do not want to work at BTL, or pretend that I am working at BTL, yet it appears that the responsibility of interpreting the standards document is a “local matter” that I must shoulder

IMHO, the BTL should have nothing whatsoever to do with establishing what tests are, how the standard should be interpreted, or what constitutes conformance. They should do tests period. BTL, like everyone else in the world, is welcome to sit at the table in the SSPC and share opinions on these matters, but only the controlled consensus process of the SSPC should be the source of interpretation and changes to the standard. That's 100% in the end-user's interest and their only defense against clique-controlled groups. That process has proven robust in this regard, which is why it is used by every international standard's making body.