INTERPRETATION IC 135-2024-11 OF
ANSI/ASHRAE STANDARD 135-2024 BACnet® -
A Data Communication Protocol for Building
Automation and Control Networks

Approval Date: June 23, 2025

Request from: Michael Osborne (btbbasconsulting@gmail.com), BTB Consulting, 408 - 9864 Fourth St, Sidney, BC, V8L 2Z4. (Phone: 250-589-1979)

Reference: This request for interpretation refers to ANSI/ASHRAE Standard 135-2024 and pertains to the requirements to accept a relinquish of non-commandable properties.

Background: The below changes (in green) were introduced in 135-2016Add *br*-2 (PR 21).

The Rational for Section br-2 is:

> The standard mandates that non-commandable objects ignore provided command priorities but makes no mention of what to do when a relinquish command comes along (i.e., a NULL is written to a non-commandable property).
>
> This results in write errors which really should not occur.
>
> This change mandates that those writes not be failed due to an invalid datatype. The NULL is ignored, but the write is reported to have succeeded.

15.9.2 Service Procedure (WriteProperty Service))

After verifying the validity of the request, the responding BACnet-user shall attempt to modify the specified property of the specified object using the value provided in the 'Property Value' parameter. If the modification attempt is successful, a 'Result(+)' primitive shall be issued. If the modification attempt fails, a 'Result(-)' primitive shall be issued indicating the reason for the failure. Interpretation of the conditional Priority parameter shall be as defined in Clause 19.2.

If an attempt is made to relinquish a property that is not commandable and for which Null is not a supported datatype, if no other error conditions exist, the property shall not be changed, and the write shall be considered successful. See Clause 19.

15.10.2 Service Procedure (WritePropertyMultiple Service)

For each 'Write Access Specification' contained in the 'List of Write Access Specifications', the value of each specified property shall be replaced by the property value provided in the 'Write Access Specification' and a 'Result(+)' primitive shall be issued, indicating that the service request was carried out in its entirety. Interpretation of the conditional Priority parameter shall be as specified in Clause 19.

If, in the process of carrying out the modification of the indicated properties in the order specified in the 'List of Write Access Specifications', a property is encountered that cannot be modified, the responding BACnet-user shall issue a 'Result(-)' response primitive indicating the reason for the failure. The result of this service shall be either that all of the specified properties or only the properties up to, but not including, the property specified in the 'First Failed Write Attempt' parameter were successfully modified.

A BACnet-Reject-PDU shall be issued only if no write operations have been successfully executed, indicating that the service request was rejected in its entirety. If any of the write operations contained in the 'List of Write Access

©2025 ASHRAE. All Rights reserved.

Specifications' have been successfully executed, a Result(-) response indicating the reason for the failure shall be issued as described above.

In the case that the 'Object Identifier', the 'Property Identifier', or the 'Property Array Index' cannot be successfully decoded after at least one write operation has completed successfully, the object instance portion of the 'Object Identifier' specified in the 'First Failed Write Attempt' shall contain the instance value 4194303. In this case, the value of the 'Property Identifier' parameter and the 'Property Array Index' parameter is a local matter.

If an attempt is made to relinquish a property that is not commandable and for which Null is not a supported datatype, if no other error conditions exist, the property shall not be changed, and the write to that property shall be considered successful. See Clause 19.

19.2.1 Prioritization Mechanism (Command Prioritization)

For BACnet objects, commands are prioritized based upon a fixed number of priorities that are assigned to command-issuing entities. A prioritized command (one that is directed at a commandable property of an object) is performed via a WriteProperty service request or a WritePropertyMultiple service request. The request primitive includes a conditional 'Priority' parameter that ranges from 1 to 16. Each commandable property of an object has an associated priority table that is represented by a Priority_Array property. The Priority_Array consists of an array of commanded values in order of decreasing priority. The first value in the array corresponds to priority 1 (highest), the second value corresponds to priority 2, and so on, to the sixteenth value that corresponds to priority 16 (lowest).

An entry in the Priority_Array may have a commanded value or a Null. A Null value indicates that there is no existing command at that priority. An object continuously monitors all entries within the priority table in order to locate the entry with the highest priority non-Null value and sets the commandable property to this value.

A commanding entity (application program, operator, etc.) may issue a command to write to the commandable property of an object, or it may relinquish a command issued earlier. Relinquishing of a command is performed by a write operation similar to the command itself, except that the commandable property value is Null. Relinquishing a command places a Null value in the Priority_Array corresponding to the appropriate priority. This prioritization approach shall be applied to local actions that change the value of commandable properties as well as to write operations via BACnet services.

If an attempt is made to write to a commandable property without explicitly specifying the priority, a default priority of 16 (the lowest priority) shall be assumed. If an attempt is made to write to a property that is not commandable with a specified priority, the priority shall be ignored. The Priority_Array property is read-only. Its values are changed indirectly by writing to the commandable property itself.

If an attempt is made to relinquish a non-commandable property and for which Null is not a supported datatype, if no other error conditions exist, the property shall not be changed, and the write to that property shall be considered successful.

Rational: The Relinquish action is meant for commandable properties and the only commandable property is Present_Value. Presently, the only objects that can contain a commandable or non-commandable Present_Value property are value objects.

Because the standard uses the term ‘relinquish’ (writing Null at a priority) it infers non-commandable Present_Value properties.

From an interoperability point of view, the changes in addendum br-2 allow a client to relinquish any Present_Value property without determining if an object is commandable or dealing with error codes. There is no interoperable benefit in requiring a device accept and ignore a relinquish to the High_Limit property.

©2025 ASHRAE. All Rights reserved.

Problem: The standard does not explicitly limit what writable non-commandable properties must accept and not write a relinquish request.

Interpretation: The standard should have restricted the requirement to quietly accept an attempt to relinquish to only those properties that are optionally commandable for instances that are not commandable.

Question: Is this Interpretation correct?

Answer:

Yes

Comment:

©2025 ASHRAE. All Rights reserved.