BACnet Errata
ANSI/ASHRAE STANDARD 135-2020
A Data Communication Protocol for Building Automation and Control Networks

April 27, 2022

This document lists all known errata to ANSI/ASHRAE Standard 135-2020 as of the above date. Each entry is cited first by clause, then page number, except where an erratum covers more than one clause. The back page marking identifying the electronic publication of Standard 135-2020 is “Product code: D-86451 9/20”.

Changes are indicated by using strikeout for text to be removed and italics for text to be added, unless noted otherwise. Grey highlighting is used for marking small corrections.


An initiating peer shall send a Heartbeat-Request message to the accepting peer if the initiating peer has not received an BVLC message over the connection within the heartbeat timeout.

On receipt of Heartbeat-Request, the accepting peer shall respond with a Heartbeat-ACK message to the initiating peer.

Although the accepting peer is not required to keep established BACnet/SC connections alive through periodically sending Heartbeat-Request messages to the initiating peer, the accepting peer may send a Heartbeat-Request message to the initiating peer at any time in order to determine whether the initiating peer and the connection is still alive.

2) Annex AB.7.4 and sub-clauses, p. 1406 - 1407: Clarify terms used for certificates.

AB.7.4 Connection Security

The establishment of a secure WebSocket connection shall be performed as defined in RFC 6455. For establishing a secure WebSocket connection, mutual TLS authentication shall be performed. "Mutual authentication" in this context means that both the initiating peer and the accepting peer shall:

(a) Validate that the peer's operational certificate is well formed.
(b) Validate that the peer's operational certificate is active as of the current date and not expired.
(c) Validate that the peer's operational certificate is not revoked, if such information is available.
(d) Validate that the peer's operational certificate is directly signed by one of the locally configured CA issuer certificates.

AB.7.4.1.1 Operational Credentials

Operational credentials include the certificate of a device, the related private key, and the accepted signing CA issuer certificates that are used to connect to a BACnet/SC network of an installation. A device may have other certificates and private keys and signing CA certificates being used for manufacturer specific communication. These credentials are not considered operational credentials and may be considered to be part of the factory default condition of the device. See Clause AB.7.4.2.

Before deployment to an active network, the connection peers shall be configured with a CA issuer certificate store containing one or more CA issuer certificates of those signing CAs that are accepted to have signed the peer's certificate, and a unique operational certificate with matching private key. The operational certificate shall be issued and directly signed by a signing CA whose CA issuer certificate is configured in the CA issuer certificate store. This allows peer-to-peer mutual authentication so that the accepting peer and the initiating peer can each verify that the certificate presented to it was signed by one of the signing CAs in its CA issuer certificate store.
AB.7.4.1.2 Signing CA

The choice of one or multiple signing CAs to sign the operational certificates used in a site shall be dictated by site policy. Each signing CA shall be controlled by the site and can be a root CA or an intermediate CA.

The signing CAs shall support processing of certificate signing requests in Privacy Enhanced Mail (PEM) format (RFC 7468) conveying a certificate signing request and return the signed certificates in PEM formatted PKCS7 structure.

AB.7.4.1.3 Configuring Operational Certificates

…

For devices that cannot generate their own public/private key pairs, the key pair needs to be generated by a configuration tool. In this case, the tool shall generate the key pair and create a certificate signing request based on certificate parameters defined by the installation. The tool shall submit the certificate signing request to the signing CA for the installation. The signed operational certificate returned from the signing CA, the private key, and the CA issuer certificates required for the installation are configured into the device by the tool. The private key shall only be transferred in a secured environment, or over communication secured by TLS.

A device that supports an internal security function that allows it to generate and store its private keys by itself is not allowed to expose the private keys, and may not be allowed to accept a private key from a configuration tool. To create a signed operational certificate, the configuration tool provides certificate parameters of the installation to the device and initiates a certificate signing request generation by the device. The certificate signing request is sent to a signing CA of the installation. The signed operational certificate returned from the signing CA, and the CA issuer certificates for the accepted CAs as required for the installation are configured into the device by the tool.

…

AB.7.4.2.1 Reset to Factory Defaults

…

Performing a reset to "factory defaults" condition shall erase all operational certificates and respective private keys and all CA issuer certificates from all BACnet/SC network ports. Any sensitive data the device contains shall also be erased. It is not allowed to simply block access to existing sensitive data while in the factory defaults condition because an attacker with physical access can use this condition to insert new operational credentials and then use that false trust relationship to access sensitive data that was not erased.

3) Annex AB.7.5.1, p. 1407: Remove undefined error code.

…

<table>
<thead>
<tr>
<th>Situation</th>
<th>Error Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>The security parameters of the client and server do not match.</td>
<td>TLS_SECURITY_PARAMETER_MISMATCH</td>
</tr>
</tbody>
</table>

…

4) Clause 12.X.Y Property_List, Use standard language for Property_List.

This read-only property, of type a BACnetARRAY of BACnetPropertyIdentifier property identifiers, contains one BACnetPropertyIdentifier property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

5) Clause 15.9.1.1.5, p. 752: Priority parameter defined as Unsigned but described as integer.

This parameter, of type Unsigned, shall be an integer in the range 1..16, which indicates the priority assigned to this write operation.
6) **Clause 15.10.3.2.4**, p. 756: Priority parameter defined as Unsigned but described as integer.

This parameter, *of type Unsigned*, shall be an integer in the range 1..16, which indicates the priority assigned to this write operation. …

7) **Clause 21.6**, p. 886: successful-actions-only parameter type incorrectly defined.

...  
**BACnetAuditLogQueryParameters** ::= CHOICE {  
by-target [0] SEQUENCE {  
target-device-identifier [0] BACnetObjectIdentifier,  
target-device-address [1] BACnetAddress OPTIONAL,  
target-object-identifier [2] BACnetObjectIdentifier OPTIONAL,  
target-property-identifier [3] BACnetPropertyIdentifier OPTIONAL,  
target-array-index [4] Unsigned OPTIONAL,  
target-priority [5] Unsigned(1..16) OPTIONAL,  
operations [6] BACnetAuditOperationFlags OPTIONAL,  
successful-actions-only [7] BOOLEAN BACnetSuccessFilter  
},  
by-source [1] SEQUENCE {  
source-device-identifier [0] BACnetObjectIdentifier,  
source-device-address [1] BACnetAddress OPTIONAL,  
source-object-identifier [2] BACnetObjectIdentifier OPTIONAL,  
operations [3] BACnetAuditOperationFlags OPTIONAL,  
successful-actions-only [4] BOOLEAN BACnetSuccessFilter  
}  
}  
...

8) **Clause 21.2.3**, p. 865: start-at-sequence-number parameter type incorrectly defined.

...  
**AuditLogQuery-Request** ::= SEQUENCE {  
audit-log [0] BACnetObjectIdentifier,  
query-parameters [1] BACnetAuditLogQueryParameters,  
start-at-sequence-number [2] Unsigned64 Unsigned32 OPTIONAL,  
requested-count [3] Unsigned16  
}  
...

9) **Table 12-47**, p. 435: Value_Source and Value_Source_Array missing footnote references.

<table>
<thead>
<tr>
<th>Property Identifier</th>
<th>Property Datatype</th>
<th>Conformance Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td>O^8,10,12</td>
</tr>
<tr>
<td>Value_Source</td>
<td>BACnetValueSource</td>
<td>O^8,11</td>
</tr>
<tr>
<td>Value_Source_Array</td>
<td>BACnetARRAY[16] of BACnetValueSource</td>
<td></td>
</tr>
</tbody>
</table>

10) **Table 12-4**, p. 179: Analog Value object Audit_Priority_Filter footnote missing Commandable conditionality.
12) Table 12-10, p. 205: Binary Value object Audit_Priority_Filter footnote missing Commandable conditionality.

<table>
<thead>
<tr>
<th>Property Identifier</th>
<th>Property Datatype</th>
<th>Conformance Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Audit_Level</td>
<td>BACnetAuditLevel</td>
<td>O\textsuperscript{12}</td>
</tr>
<tr>
<td>Auditable_Operations</td>
<td>BACnetAuditOperationFlags</td>
<td>O\textsuperscript{12}</td>
</tr>
<tr>
<td>Audit_Priority_Filter</td>
<td>BACnetOptionalPriorityFilter</td>
<td>O\textsuperscript{12/13}</td>
</tr>
</tbody>
</table>

13) Annex AB.6.2.3, p. 1405: Disconnecting-ACK message should be from the initiating peer.

... In state DISCONNECTING

Disconnect-ACK received

On receipt of a Disconnect-ACK message from the \textit{initiating} accepting peer, close the WebSocket connection, and enter the IDLE state.

...
may find it convenient to have all of their Network Port objects support the same list of properties regardless of Network_Type. *If a property is present but not applicable to the specified Network_Type, then its content is a local matter. This is permitted, but not required.*

... 

15) **Clause 12.56.9,** p. 549: Clarify Protocol_Level of PROTOCOL includes BACnet and non-BACnet protocols.

This property, of type BACnetProtocolLevel, indicates whether the object represents a physical network interface (PHYSICAL), a BACnet or non-BACnet protocol (PROTOCOL), the BACnet use of the protocol (BACNET_APPLICATION), or a non-BACnet use of the protocol (NON_BACNET_APPLICATION).

16) **Clause 12.56.10,** p. 549: Clarify Protocol_Level property levels in the Reference_Port property.

... 

If this property has a value of 4194303, then this object has not been assigned a lower protocol layer. If the object is capable of representing all protocol layers in a single object, then this is a valid configuration and the object shall behave as if this property were absent. If the object is not capable of representing all protocol layers in a single object, *and Protocol_Level is not PHYSICAL*, then this is an indication that the object is not yet configured.

*If Protocol_Level has a value of PHYSICAL, then this property shall have a value of 4194303.*

---

**Figure 12-18.** Example Network Port With No Hierarchy Chain

A Network Port object is misconfigured if the referenced Network Port object has a Protocol_Level of BACNET_APPLICATION, a Protocol_Level of NON_BACNETAPPLICATION, or the referenced Network Port object does not exist.

...
12.56.10.1 Network Port Hierarchies

Support for Network Port object hierarchies is optional.

In the normal case, a single hierarchy chain consists of a Network Port object with a Protocol_Level of PHYSICAL at the bottom; one or more Network Port objects with their Protocol_Level set to PROTOCOL, and a Network Port object with a Protocol_Level of BACNET_APPLICATION or NON_BACNET_APPLICATION at the top. Multiple Network Port objects can reference a PROTOCOL or PHYSICAL Network Port object.

A Network Port object with a Protocol_Level of BACNET_APPLICATION, NON_BACNET_APPLICATION, or PHYSICAL shall not be in the middle of a hierarchy chain.

17) Clause 9.3.10, p. 102: The BACnet Extended Data Not Expecting Reply frame type is not limited to master nodes.

This COBS-encoded frame is used by master nodes to convey the data parameter of a DL_UNITDATA.request whose data_expecting_reply parameter is FALSE and whose data parameter length is between 502 and 1497 octets, inclusive.

18) Clause 12.56.34, p. 558: Missing BACnet_IPv6_Mode in BBMD_Broadcast_Distribution_Table.

This property, of type BACnetLIST of BACnetBDTEntry, is required to be present and writable if BACnet_IP_Mode or BACnet_IPv6_Mode is BBMD.


This property, of type BOOLEAN, indicates whether (TRUE) or not (FALSE) this device shall accept foreign device registrations. This property is required to be present and writable if BACnet_IP_Mode or BACnet_IPv6_Mode is BBMD.

20) Clause 12.56.37, p. 559: Missing BACnet_IPv6_Mode in FD_BBMD_Address.

This property, of type BACnetHostNPort, indicates the BBMD with which the local device is to register as a foreign device when BACnet_IP_Mode or BACnet_IPv6_Mode is FOREIGN. This property shall be present and writable if BACnet_IP_Mode or BACnet_IPv6_Mode is FOREIGN.


This property, of type Unsigned16, indicates the Time-To-Live value, in seconds, to be used in the Register-Foreign-Device BVLL message. This property shall be present and writable if BACnet_IP_Mode or BACnet_IPv6_Mode is FOREIGN.

22) Table 12-64, p. 519: Lighting Output object, Notification-Class property, incorrect Property Datatype.

<table>
<thead>
<tr>
<th>Property Identifier</th>
<th>Property Datatype</th>
<th>Conformance Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification_Class</td>
<td>Unsigned16</td>
<td>O1,4</td>
</tr>
<tr>
<td>…</td>
<td>BACnetARRAY[3]</td>
<td></td>
</tr>
<tr>
<td>…</td>
<td>BACnetTimeStamp</td>
<td></td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td></td>
</tr>
</tbody>
</table>
23) **Table 12-71**, p. 543: Network Port object, Link_Speed property, incorrect Conformance Code. (IC-135-2020-5.pdf)

<table>
<thead>
<tr>
<th>Property Identifier</th>
<th>Property Datatype</th>
<th>Conformance Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link_Speed</td>
<td>Real</td>
<td>RO</td>
</tr>
</tbody>
</table>


**K.6.7 BIBB - Network Management-Foreign Device Registration-A (NM-FDR-A)**

Devices claiming conformance to this BIBB shall be able to register as foreign device.

- Supports, at a minimum, foreign device registration lifetimes in the range 30 seconds – 9 hours.

<table>
<thead>
<tr>
<th>BACnet Virtual Link Layer Message</th>
<th>Initiate</th>
<th>Execute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register-Foreign-Device</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Distribute Broadcast-To-Network</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Forwarded-NPDU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original-Unicast-NPDU</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

25) **Table 12-13**, p. 221: Incorrect footnote reference for Audit_Level and Auditable_Operations (Error importing from Addendum bi).

<table>
<thead>
<tr>
<th>Property Identifier</th>
<th>Property Datatype</th>
<th>Conformance Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audit_Notification_Recipient</td>
<td>BACnetRecipient</td>
<td>O^{19,20}</td>
</tr>
<tr>
<td>Audit_Level</td>
<td>BACnetAuditLevel</td>
<td>O^{20/21}</td>
</tr>
<tr>
<td>Auditable_Operations</td>
<td>BACnetAuditOperationFlags</td>
<td>O^{20/21}</td>
</tr>
<tr>
<td>Device_UUID</td>
<td>OCTET STRING (Size(16))</td>
<td>O^{21/22}</td>
</tr>
<tr>
<td>Tags</td>
<td>BACnetARRAY[N] of BACnetNameValue</td>
<td>O^{22}</td>
</tr>
</tbody>
</table>

19. This property is required to be writable if present.
20. This property shall be present if, and only if, the device supports audit reporting.
21. *This property shall be present only if the device supports audit reporting.*
22. This property shall be present if the device supports BACnet/SC network ports.

26) **Clause 12.56.8**, p. 548: Incorrect reference clause for Network_Type of VIRTUAL.

This property shall have one of the following values:

- ARCNET
- IPV4
- IPV6
- ETHERNET
- LONTALK
- MSTP
- MS/TP, as defined in Clause 9.
PTP  Point-To-Point, as defined in Clause 10.
SERIAL  A physical serial port.
ZIGBEE  ZigBee as defined in Annex O
VIRTUAL Indicates that this port represents the configuration and properties of a virtual network as described in Clause H.2 Clauses H.1.1.1 and H.1.1.2.

<Proprietary Enum Values> A vendor may use other proprietary enumeration values to indicate that this port represents the use of message structures, procedures, and medium access control techniques other than those contained in this standard. For proprietary extensions of this enumeration, see Clause 23.1 of this standard.


<table>
<thead>
<tr>
<th>Property Identifier</th>
<th>Property Datatype</th>
<th>Conformance Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification_Class</td>
<td>Unsigned</td>
<td>R^O^1,2</td>
</tr>
</tbody>
</table>

1 These properties are required if the object supports intrinsic reporting.
2 These properties shall be present only if the object supports intrinsic reporting.
3 If this property is present, then the Reliability property shall be present.
4 This property shall be present only if the device supports audit reporting.

28) Clause 12, throughout. Incorrect reference clause for Acked_Transitions property. References 13.2.2.15, should reference 13.2.3.

This read-only property, of type BACnetEventTransitionBits, shall convey three flags that separately indicate the acknowledgment state for TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events (see Clause 13.2.2.1.5 13.2.3). Each flag shall have the value TRUE if no event of that type has ever occurred for the object.

29) Clause K.8.6, p 1116: Incorrect reference to K-30, should be K-2

The A device is able to use ReadProperty to retrieve and present all standard properties of the Audit Reporter and Audit Log object types, except those listed in Table K-302.

31) Addendum 135-2020cc Clause 21, p. 798