



**BSR/ASHRAE Addendum *cf* to
ANSI/ASHRAE Standard 135-2020**

Public Review Draft

Proposed Addendum *cf* to Standard 135-2020, BACnet[®] - A Data Communication Protocol for Building Automation and Control Networks

**First Public Review (June 2022)
(Draft shows Proposed Changes to Current Standard)**

This draft has been recommended for public review by the responsible project committee. To submit a comment on this proposed standard, go to the ASHRAE website at www.ashrae.org/standards-research--technology/public-review-drafts and access the online comment database. The draft is subject to modification until it is approved for publication by the Board of Directors and ANSI. Until this time, the current edition of the standard (as modified by any published addenda on the ASHRAE website) remains in effect. The current edition of any standard may be purchased from the ASHRAE Online Store at www.ashrae.org/bookstore or by calling 404-636-8400 or 1-800-727-4723 (for orders in the U.S. or Canada).

This standard is under continuous maintenance. To propose a change to the current standard, use the change submittal form available on the ASHRAE website, www.ashrae.org.

The appearance of any technical data or editorial material in this public review document does not constitute endorsement, warranty, or guaranty by ASHARE of any product, service, process, procedure, or design, and ASHRAE expressly disclaims such.

© 2022 ASHRAE. This draft is covered under ASHRAE copyright. Permission to reproduce or redistribute all or any part of this document must be obtained from the ASHRAE Manager of Standards, 180 Technology Parkway NW, Peachtree Corners, GA 30092. Phone: 404-636-8400, Ext. 1125. Fax: 404-321-5478. E-mail: standards.section@ashrae.org.

ASHRAE, 180 Technology Parkway NW, Peachtree Corners, GA 20092

[This foreword, the table of contents, the introduction, and the “rationales” on the following pages are not part of this standard. They are merely informative and do not contain requirements necessary for conformance to the standard.]

FOREWORD

The purpose of this addendum is to present a proposed change for public review. These modifications are the result of change proposals made pursuant to the ASHRAE continuous maintenance procedures and of deliberations within Standing Standard Project Committee 135. The proposed changes are summarized below.

135-2020*cf*-1 Formal Definition of the 'data_attributes' Parameter, p.3.

135-2020*cf*-2 Redefinition of 'Must Understand' for data options, p.4.

135-2020*cf*-3 Changes to segmentation to enforce data attribute consistency, p.6.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2020 is indicated through the use of *italics*, while deletions are indicated by ~~strike through~~. Where entirely new subclauses are proposed to be added, plain type is used throughout. Only this new and deleted text is open to comment at this time. All other material in this document is provided for context only and is not open for public review comment except as it relates to the proposed changes.

The use of placeholders like XX, YY, ZZ, X1, X2, NN, x, n, ? etc. should not be interpreted as literal values of the final published version. These placeholders will be assigned actual numbers/letters only after final publication approval of the addendum.

135-2020*cf*-1 Formal Definition of the 'data_attributes' Parameter

Rationale

The format of the 'data_attributes' parameter defined in Clauses 5.1 and 6.1 is left as a "local matter". However, Annex AB makes requirements that BACnet/SC 'Data Options' are to be included in the 'data_attributes' parameter. The 'Data Options' have a very specific format, and each option has a defined type, including the option for proprietary types. This creates an *implied* requirement that this information is retained in the 'data_attributes' parameter even if the exact internal format is still unspecified.

This change formalizes the definition of the *information* that each attribute contains and adds a new piece of information to fix the segmentation issues that are addressed by Section 3.

[Add new **Clause 5.1.X**, p. 30]

5.1.X Data Attributes

The ICI primitives defined in Clause 5.1 include the extensible parameter named 'data_attributes' that is a logical collection of individual attributes that accompany an NPDU or APDU. Each attribute contains the following information:

- a) an optional vendor-identifier
- b) the attribute-type
- c) the attribute-data
- d) an every-segment flag

The vendor-identifier is an unsigned number in the range 1-65535. If it is absent, then the attribute has an attribute-type defined by this standard. If the vendor-identifier is present, then the attribute has a proprietary attribute-type.

The attribute-type is an unsigned number. For standard types, the number is assigned elsewhere in this standard and has a range of 0-31. For proprietary attributes, the number is assigned by the organization identified by the vendor-identifier and has a range of 0-255.

The attribute-data is an ordered sequence of octets. Its meaning is specific to the attribute-type. Its maximum length is 4096 octets.

The every-segment flag is a Boolean indicating that, if TRUE, this attribute shall be sent with every segment of a segmented message, and if FALSE, it shall only be sent with the first segment.

Attributes of the same attribute-type can be repeated in the 'data_attributes' collection. The presence of a repeated attribute with the same value might be significant, therefore, implementations shall not remove duplicates.

The internal format of this information in the ICI primitives is a local matter. However, datalinks that convey 'data_attributes' externally will specify explicitly how each of these information items is encoded. For example, see BACnet/SC, Clause AB.2.3.

135-2020*cf*-2 Redefinition of 'Must Understand' for data options

Rationale

This change repurposes the 'Must Understand' bit for 'Data Options' to be the first/every segment indicator.

The only data option in the standard currently is 'Secure Path' which must be sent with every segment. Therefore, the bit is renamed to 'Every Segment' since 'Secure Path' already has this bit set to 1.

This change does not affect the meaning of the 'Must Understand' bit for 'Destination Options'.

[Change **Clause AB.2.3**, p. 1385]

AB.2.3 Header Options

...

The 'Header Marker' octet for 'Destination Options' includes the fields as follows:

Bit 7	More Options	1 = Another header option follows in the current header option list. 0 = This is the last header option in the current header option list.
Bit 6:	Must Understand	1 = This header option must be understood for consuming the message. 0 = This header option can be ignored if not understood.
Bit 5:	Header Data Flag	1 = The 'Header Length' and 'Header Data' fields are present 0 = The 'Header Length' and 'Header Data' fields are absent
Bits 4..0:	Header Option Type	1..31, The numeric header option type.

The 'Header Marker' octet for 'Data Options' includes the fields as follows:

Bit 7	More Options	1 = Another header option follows in the current header option list. 0 = This is the last header option in the current header option list.
Bit 6:	Every Segment	1 = This header option shall be sent with every segment. 0 = This header option shall be sent with the first, or only, segment and shall not be sent with subsequent segments.
Bit 5:	Header Data Flag	1 = The 'Header Length' and 'Header Data' fields are present 0 = The 'Header Length' and 'Header Data' fields are absent
Bits 4..0:	Header Option Type	1..31, The numeric header option type.

The 'More Options' flag indicates if the header option is the last option in the current header options list (0), or at least one more header option follows in the current header options list (1).

For the handling of the 'Must Understand' flag and the processing of header options when sending, forwarding, broadcasting, or receiving BVLC messages with ~~header~~ destination options, see Clause AB.3.1.4.

For the handling of the 'Every Segment' flag and the processing of header options when sending or receiving NPDUs with data options, see Clause 5.2.1.1 and the subclauses of Clause 5.4.

...

[Change **Clause AB.2.3.1**, p. 1386]

AB.2.3.1 Secure Path Header Option

The 'Secure Path' header option specifies, by its presence, whether the service being requested represents a message which has only been transferred by BACnet/SC data links and secure connect BACnet routers.

The 'Secure Path' header option consists of the following fields.

Header Marker	1-octet	'Last Option' = 0 or 1, 'Must Understand' 'Every Segment' = 1, 'Header Data Flag' = 0, 'Header Option Type' = 1
---------------	---------	---

This header option, if present, shall be a data option in the 'Data Options' parameter

135-2020*cf*-3 Changes to segmentation to enforce data attribute consistency

Rationale

There are two kinds of data attributes: those that are designated as "every segment" and those that are "first segment". This change enforces the rule that the "first segment" attributes can only be sent with the first segment, and that the "every segment" attributes must be sent with every segment and must be consistent in presence and value.

[Change **Clause 5.2.1.1**, p. 20]

5.2.1.1 Rules for Segmenting APDU Data Streams

Each BACnet message is encoded into a sequence of tags and values according to the relevant ASN.1 definitions in Clause 21 and the encoding rules of Clause 20. The following rules apply to segmenting this data stream:

- (a) If possible, an entire message shall be sent in a single APDU.
- (b) If an entire message cannot be sent in a single APDU, the message shall be segmented into the minimum number of APDUs possible.
- (c) Messages shall be segmented only at octet boundaries.
- (d) *If data attributes are included, the attributes designated as "every segment" shall be present and have the same value for every segment, and attributes designated "first segment" shall only be present with the first segment.*

[Change **Clause 5.4.3**, p. 27]

[note to reviewer, FillWindow is called for segments 2..N so it does not apply to the first segment]

5.4.3 Function FillWindow

The function "FillWindow" sends PDU segments either until the window is full or until the last segment of a message has been sent. No more than T_{seg} may be allowed to elapse between the receipt of a SegmentACK APDU and the transmission of a segment. No more than T_{seg} may be allowed to elapse between the transmission of successive segments of a sequence.

function FillWindow(sequenceNumber)

- (1) Set local variable ix to zero.
- (2) If the next segment to transmit (the segment numbered sequenceNumber plus ix) is the final segment, goto step (7).
- (3) Issue an N-UNITDATA.request with 'data_expecting_reply' = TRUE to transmit the next BACnet APDU segment *along with all data attributes that are designated as "every segment"*, with 'segmented-message' = TRUE, 'more-follows' = TRUE, 'proposed-window-size' equal to ProposedWindowSize, and 'sequence-number' = sequenceNumber plus ix, modulo 256.
- (4) Set ix equal to ix plus one.
- (5) If ix is less than ActualWindowSize, goto step (2).
- (6) Goto step (9).
- (7) Issue an N-UNITDATA.request with 'data_expecting_reply' = TRUE to transmit the final BACnet APDU segment *along with all data attributes that are designated as "every segment"*, with 'segmented-message' = TRUE, 'more-follows' = FALSE, 'proposed-window-size' = ProposedWindowSize, and 'sequence-number' = sequenceNumber plus ix, modulo 256.
- (8) Set SentAllSegments to TRUE, indicating that all segments have been transmitted at least once.
- (9) Return to the caller.

[Change **Clause 5.4.4**, p. 28]

5.4.4 State Machine for Requesting BACnet Users (client)

5.4.4.1 IDLE

In the IDLE state, the device waits for the local application program to request a service.

...

SendConfirmedSegmented

If CONF_SERV.request is received from the local application program and the length of the APDU is greater than maximum-transmittable-length as determined according to Clause 5.2.1, and the Max_Segments_Accepted property of the destination's Device object is not known, or Max_Segments_Accepted is known and the total APDU can be transmitted without exceeding the maximum number of segments accepted,

then assign an 'invoke-id' to this transaction; set SentAllSegments to FALSE; set RetryCount to zero; set SegmentRetryCount to zero; set InitialSequenceNumber to zero; set ProposedWindowSize to whatever value is desired; set ActualWindowSize to 1; start SegmentTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = TRUE to transmit a BACnet-Confirmed-Request-PDU containing the first segment of the message *and all of the data attributes*, with 'segmented-message' = TRUE, 'more-follows' = TRUE, 'sequence-number' = zero, and 'proposed-window-size' = ProposedWindowSize; and enter the SEGMENTED_REQUEST state to await an acknowledgment. (The method used to determine ProposedWindowSize is a local matter, except that the value shall be in the range 1 to 127, inclusive.)

...

5.4.4.2 SEGMENTED_REQUEST

In the SEGMENTED_REQUEST state, the device waits for a BACnet-SegmentACK-PDU for one or more segments of a BACnet-Confirmed-Request-PDU.

...

SegmentedComplexACK_Received

If a BACnet-ComplexACK-PDU that has sufficient security parameters is received from the network layer whose 'segmented-message' parameter is TRUE and whose 'sequence-number' parameter is zero and this device supports segmentation and SentAllSegments is TRUE,

then save the BACnet-ComplexACK-PDU segment *and all its data attributes*; stop SegmentTimer; compute ActualWindowSize based on the 'proposed-window-size' parameter of the received BACnet-ComplexACK-PDU and on local conditions; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-SegmentACK-PDU with 'negative-ack' = FALSE, 'server' = FALSE, and 'actual-window-size' = ActualWindowSize; start SegmentTimer; set LastSequenceNumber to zero; set InitialSequenceNumber to zero; set DuplicateCount to zero; and enter the SEGMENTED_CONF state to receive the remaining segments. (The method used to determine ActualWindowSize is a local matter, except that the value shall be less than or equal to the 'proposed-window-size' parameter of the received BACnet-ComplexACK-PDU and shall be in the range 1 to 127, inclusive.)

...

5.4.4.2 AWAIT_CONFIRMATION

In the AWAIT_CONFIRMATION state, the device waits for a response to a BACnet-Confirmed-Request-PDU.

...

SegmentedComplexACK_Received

If a BACnet-ComplexACK-PDU that has sufficient security parameters is received from the network layer whose 'segmented-message' parameter is TRUE and whose 'sequence-number' parameter is zero and this device supports segmentation,

[note to reviewer: the "save the BACnet-ComplexACK-PDU" was implied and required but not stated explicitly]
then *save the BACnet-ComplexACK-PDU segment and all its data attributes*; stop RequestTimer; compute ActualWindowSize based on the 'proposed-window-size' parameter of the received BACnet-ComplexACK-PDU and on local conditions; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-SegmentACK-PDU with 'negative-ack' = FALSE, 'server' = FALSE, and 'actual-window-size' = ActualWindowSize; start SegmentTimer; set LastSequenceNumber to zero; set InitialSequenceNumber to zero; set DuplicateCount to zero; and enter the SEGMENTED_CONF state to receive the remaining segments. (The method used to determine ActualWindowSize is a local matter, except that the value shall be less than or equal to the 'proposed-window-size' parameter of the received BACnet-ComplexACK-PDU and shall be in the range 1 to 127, inclusive.)

...

5.4.4.2 SEGMENTED_CONF

In the SEGMENTED_CONF state, the device waits for one or more segments in response to a BACnet-SegmentACK-PDU.

...

[Insert new transition between "NewSegmentReceived_NoSpace" and "NewSegmentReceived"]

[Note to reviewer: the words "has sufficient security parameters" is not new language here. It is present on every transition and will be clarified in a separate addendum]

NewSegmentReceived_InconsistentAttributes

If a BACnet-ComplexACK-PDU that has sufficient security parameters is received from the network layer whose 'segmented-message' parameter is TRUE; whose data attributes include any attributes that are designated as "first segment" or whose data attributes that are designated "every segment" are inconsistent in presence and value with those sent with the first segment,

then stop SegmentTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = FALSE and 'abort-reason' = INCONSISTENT_ATTRIBUTES; send ABORT.indication with 'server' = FALSE and 'abort-reason' = INCONSISTENT_ATTRIBUTES to the local application program; and enter the IDLE state.

[Change **Clause 5.4.5**, p. 35]

5.4.5 State Machine for Responding BACnet Users (servers)

5.4.5.1 IDLE

In the IDLE state, the device waits for a PDU from the network layer.

...

ConfirmedSegmentedReceived

If a BACnet-Confirmed-Request-PDU whose 'segmented-message' parameter is TRUE, whose 'sequence-number' parameter is zero, and whose 'proposed-window-size' is greater than zero and less than or equal to 127 is received from the network layer and the local device supports the reception of segmented messages,

then save the BACnet-Confirmed-Request-PDU segment *and all its data attributes*; compute ActualWindowSize based on the 'proposed-window-size' parameter of the received BACnet-Confirmed-Request-PDU and on local conditions; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-SegmentACK-PDU with 'negative-ack' = FALSE, 'server' = TRUE, and 'actual-window-size' = ActualWindowSize; start SegmentTimer; set LastSequenceNumber to zero; set InitialSequenceNumber to zero; set DuplicateCount to zero; and enter the SEGMENTED_REQUEST state to receive the remaining segments. (The method used to determine ActualWindowSize is a local matter, except that the value shall be less than or equal to the 'proposed-window-size' parameter of the received BACnet-Confirmed-Request-PDU and shall be in the range 1 to 127, inclusive.)

...

5.4.5.2 SEGMENTED_REQUEST

In the SEGMENTED_REQUEST state, the device waits for segments of a BACnet-Confirmed-Request-PDU.

[Insert new transition before "NewSegmentReceived"]

[Note to reviewer: the language "secured with the same settings" is existing language used throughout this clause and is not related to this change]

NewSegmentReceived_InconsistentAttributes

If a BACnet-Confirmed-Request-PDU that is secured with the same settings as the original PDU is received from the network layer whose 'segmented-message' parameter is TRUE; whose data attributes include any attributes that are designated as "first segment" or whose data attributes that are designated "every segment" are inconsistent in presence and value with those sent with the first segment,

then stop SegmentTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = TRUE and 'abort-reason' = INCONSISTENT_ATTRIBUTES; and enter the IDLE state.

...

5.4.5.2 AWAIT_RESPONSE

In the AWAIT_RESPONSE state, the device waits for the local application program to respond to a BACnet-Confirmed-Request-PDU. See Clause 9.8 for specific considerations in MS/TP networks.

...

SendSegmentedComplexACK

If a CONF_SERV.response(+) is received from the local application program that is to be conveyed via a BACnet-ComplexACK-PDU, and the length of the APDU is greater than maximum-transmittable-length as determined according to Clause 5.2.1, and the device supports the transmission of segmented messages, and the client will accept a segmented response ('segmented-response-accepted' parameter in BACnet-ConfirmedRequest-PDU is TRUE),

then set SegmentRetryCount to zero; set InitialSequenceNumber to zero; set ProposedWindowSize to whatever value is desired; set ActualWindowSize to 1; start SegmentTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = TRUE to transmit a BACnet-ComplexACK-PDU containing the first segment of the message and *all of the data attributes*, with

First Public Review

'segmented-message' = TRUE, 'more-follows' = TRUE, 'sequence-number' = zero, and 'proposed-window-size' = ProposedWindowSize; and enter the SEGMENTED_RESPONSE state to await an acknowledgment.

[Change **Clause 18.10**, p. 800]

18.10 Abort Reason

...

INCONSISTENT_ATTRIBUTES - *The data attributes are not consistent between segments of a segmented message.*

...

[Change **Clause 21.4**, p 861]

18.10 Error Productions

BACnetAbortReason ::= ENUMERATED {

...

apdu-too-long (11),

inconsistent-attributes (*n*),

...

}

-- Enumerated values 0-63 are reserved for definition by ASHRAE. Enumerated values 64-255

-- may be used by others subject to the procedures and constraints described in Clause 23.

[Add a new entry to **History of Revisions**, p. 1429]

(This History of Revisions is not part of this standard. It is merely informative and does not contain requirements necessary for conformance to the standard.)

3 HISTORY OF REVISIONS

...
1	X	Addendum <i>cf</i> to ANSI/ASHRAE 135-2020 Approved by the ASHRAE Standards Committee MONTH X, 20XX; by the ASHRAE Board of Directors MONTH X, 20XX; and by the American National Standards Institute MONTH X, 20XX. <ol style="list-style-type: none">1. Formal Definition of the 'data_attributes' Paramete2. Redefinition of 'Must Understand' for data options3. Changes to segmentation to enforce data attribute consistency