



# ADDENDA

**ANSI/ASHRAE Addendum br to  
ANSI/ASHRAE Standard 135-2016**

# **A Data Communication Protocol for Building Automation and Control Networks**

Approved by ASHRAE and by the American National Standards Institute on August 26, 2019.

This addendum was approved by a Standing Standard Project Committee (SSPC) for which the Standards Committee has established a documented program for regular publication of addenda or revisions, including procedures for timely, documented, consensus action on requests for change to any part of the standard. Instructions for how to submit a change can be found on the ASHRAE® website (<https://www.ashrae.org/continuous-maintenance>).

The latest edition of an ASHRAE Standard may be purchased on the ASHRAE website ([www.ashrae.org](http://www.ashrae.org)) or from ASHRAE Customer Service, 1791 Tullie Circle, NE, Atlanta, GA 30329-2305. E-mail: [orders@ashrae.org](mailto:orders@ashrae.org). Fax: 678-539-2129. Telephone: 404-636-8400 (worldwide), or toll free 1-800-527-4723 (for orders in US and Canada). For reprint permission, go to [www.ashrae.org/permissions](http://www.ashrae.org/permissions).

© 2019 ASHRAE

ISSN 1041-2336



**ASHRAE Standing Standard Project Committee 135**  
**Cognizant TC: 1.4, Control Theory and Application**  
**SPLS Liaison: Drury B. Crawley**

Michael Osborne*, <i>Chair</i>	Bernhard Isler	David Ritter*
Coleman L. Brumley, Jr., <i>Vice-Chair</i>	Daniel Kollodge	David Robin*
Scott Ziegenfus*, <i>Secretary</i>	Jake Kopocis*	Frank Schubert
Sunil Barot	Thomas Kurowski*	Matthew Schwartz*
James F. Butler	Edward J. Macey-MacLeod*	Marcelo Richter da Silva
Salvatore Cataldi*	Frank V. Neher	Steve Sywak*
Clifford H. Copass	Carl Neilson	David B. Thompson
Brandon M. DuPrey*	H. Michael Newman*	Grant N. Wichenko*
David Fisher	Duffy O'Craven*	Teresa Zotti*
David G. Holmberg*	Jonathan Rigsby	

\* Denotes members of voting status when the document was approved for publication

---

**ASHRAE STANDARDS COMMITTEE 2019–2020**

Wayne H. Stoppelmoor, Jr., <i>Chair</i>	Susanna S. Hanson	Lawrence J. Schoen
Drury B. Crawley, <i>Vice-Chair</i>	Rick M. Heiden	Steven C. Sill
Els Baert	Jonathan Humble	Richard T. Swierczyna
Charles S. Barnaby	Srinivas Katipamula	Christian R. Taber
Niels Bidstrup	Essam E. Khalil	Russell C. Tharp
Robert B. Burkhead	Kwang Woo Kim	Adrienne G. Thomle
Thomas E. Cappellin	Larry Kouma	Michael W. Woodford
Douglas D. Fick	Cesar L. Lim	Craig P. Wray
Michael W. Gallagher	Karl L. Peterman	Jaap Hogeling, <i>BOD ExO</i>
Walter T. Grondzik	Erick A. Phelps	Malcolm D. Knight, <i>CO</i>

Steven C. Ferguson, *Senior Manager of Standards*

---

**SPECIAL NOTE**

This American National Standard (ANS) is a national voluntary consensus Standard developed under the auspices of ASHRAE. *Consensus* is defined by the American National Standards Institute (ANSI), of which ASHRAE is a member and which has approved this Standard as an ANS, as "substantial agreement reached by directly and materially affected interest categories. This signifies the concurrence of more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that an effort be made toward their resolution." Compliance with this Standard is voluntary until and unless a legal jurisdiction makes compliance mandatory through legislation.

ASHRAE obtains consensus through participation of its national and international members, associated societies, and public review.

ASHRAE Standards are prepared by a Project Committee appointed specifically for the purpose of writing the Standard. The Project Committee Chair and Vice-Chair must be members of ASHRAE; while other committee members may or may not be ASHRAE members, all must be technically qualified in the subject area of the Standard. Every effort is made to balance the concerned interests on all Project Committees.

The Senior Manager of Standards of ASHRAE should be contacted for

- interpretation of the contents of this Standard,
- participation in the next review of the Standard,
- offering constructive criticism for improving the Standard, or
- permission to reprint portions of the Standard.

**DISCLAIMER**

ASHRAE uses its best efforts to promulgate Standards and Guidelines for the benefit of the public in light of available information and accepted industry practices. However, ASHRAE does not guarantee, certify, or assure the safety or performance of any products, components, or systems tested, installed, or operated in accordance with ASHRAE's Standards or Guidelines or that any tests conducted under its Standards or Guidelines will be nonhazardous or free from risk.

**ASHRAE INDUSTRIAL ADVERTISING POLICY ON STANDARDS**

ASHRAE Standards and Guidelines are established to assist industry and the public by offering a uniform method of testing for rating purposes, by suggesting safe practices in designing and installing equipment, by providing proper definitions of this equipment, and by providing other information that may serve to guide the industry. The creation of ASHRAE Standards and Guidelines is determined by the need for them, and conformance to them is completely voluntary.

In referring to this Standard or Guideline and in marking of equipment and in advertising, no claim shall be made, either stated or implied, that the product has been approved by ASHRAE.

**[This foreword 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.*

- 135-2016br-1. Add new engineering units, p. 3.**
- 135-2016br-2. Add mandate to accept writes of NULL to non-commandable properties, p. 4.**
- 135-2016br-3. Add intrinsic fault reporting to Lighting Output object type, p. 6.**
- 135-2016br-4. Deprecate Time form of timestamps, p. 10**
- 135-2016br-5. Clarify the Multi-state object types when Number\_Of\_States shrinks, p. 12**
- 135-2016br-6. Fix the language for event type and message text parameters of event notifications, p. 14**
- 135-2016br-7. Clarify the object instance 4194303, p. 16**
- 135-2016br-8. Wildcard instance for Network Port objects in ReadPropertyMultiple requests, p. 17**
- 135-2016br-9. Clarify the timestamp of trend log and trend log multiple log records, p. 18**

In the following document, language to be added to existing clauses of ANSI/ASHRAE Standard 135-2016 and Addenda 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 addendum 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.

### 135-2016br-1. Add new engineering units.

#### Rationale

Engineering units related to beer brewing measurement are added. There are BACnet applications for automated monitoring of beer brewing which can utilize these new engineering units.

[Change **Clause 21**, p. 814]

**BACnetEngineeringUnits** ::= ENUMERATED { -- See below for numerical order

...

--Other

...

<i>degrees-lovibond</i>	(47816),
<i>alcohol-by-volume</i>	(47817),
<i>international-bittering-units</i>	(47818),
<i>europaean-bitterness-units</i>	(47819),
<i>degrees-plato</i>	(47820),
<i>specific-gravity</i>	(47821),
<i>europaean-brewing-convention</i>	(47822)

...

-- Numerical Order Reference

...

-- see <i>degrees-lovibond</i>	(47816),
-- see <i>alcohol-by-volume</i>	(47817),
-- see <i>international-bittering-units</i>	(47818),
-- see <i>europaean-bitterness-units</i>	(47819),
-- see <i>degrees-plato</i>	(47820),
-- see <i>specific-gravity</i>	(47821),
--see <i>europaean-brewing-convention</i>	(47822)

...

## 135-2016br-2. Add mandate to accept writes of NULL to non-commandable properties.

### Rationale

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.

[Change **Clause 15.9.2**, p. 700]

### 15.9.2 Service Procedure

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.

*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.*

[Change **Clause 15.10.2**, p. 562]

### 15.10.2 Service Procedure

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

*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.*

[Change **Clause 19.2.1**, p. 749]

### **19.2.1 Prioritization Mechanism**

...

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.*

**135-2016br-3. Add intrinsic fault reporting to Lighting Output object type.**

**Rationale**

The Lighting Output object type does not support intrinsic fault reporting.

This change makes the necessary modifications to add this functionality to the Lighting Output object type.

[Change **Clause 12.54**, p. 496]

**12.54 Lighting Output Object Type**

...

*Lighting Output objects may optionally support intrinsic reporting to facilitate the reporting of fault conditions. Lighting Output objects that support intrinsic reporting shall apply the NONE event algorithm.*

The object and its properties are summarized in Table 12-64 and described in detail in this clause.

**Table 12-64.** Properties of the Lighting Output Object Type

Property Identifier	Property Datatype	Conformance Code
...		
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	O <sup>1</sup>
Reliability	BACnetReliability	O
...		
Instantaneous_Power	REAL	O
Min_Actual_Value	REAL	O <sup>4,2</sup>
Max_Actual_Value	REAL	O <sup>4,2</sup>
Lighting_Command_Default_Priority	Unsigned	R
COV_Increment	REAL	O <sup>2,3</sup>
Event_Detection_Enable	BOOLEAN	O <sup>4,5</sup>
Notification_Class	Unsigned	O <sup>4,5</sup>
Event_Enable	BACnetEventTransitionBits	O <sup>4,5</sup>
Acked_Transitions	BACnetEventTransitionBits	O <sup>4,5</sup>
Notify_Type	BOOLEAN	O <sup>4,5</sup>
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	O <sup>4,5</sup>
Event_Message_Texts	BACnetARRAY[3] of CharacterString	O <sup>5</sup>
Event_Message_Text_Config	BACnetARRAY[3] of CharacterString	O <sup>5</sup>
Reliability_Evaluation_Inhibit	BOOLEAN	O <sup>3,6</sup>
Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R
Current_Command_Priority	BACnetOptionalUnsigned	R
Value_Source	BACnetValueSource	O <sup>4,6,8,7,9,11</sup>
Value_Source_Array	BACnetARRAY[16] of BACnetValueSource	O <sup>5,7-8,10</sup>
Last_Command_Time	BACnetTimeStamp	O <sup>5,7-8,10</sup>
Command_Time_Array	BACnetARRAY[16] of BACnetTimeStamp	O <sup>7,10</sup>
Tags	BACnetARRAY[N] of BACnetNameValue	O
Profile_Location	CharacterString	O
Profile_Name	CharacterString	O

- <sup>1</sup> *These properties are required if the object supports intrinsic reporting.*
- <sup>2,2</sup> *If either of these properties is present, they shall both be present, and they shall be writable.*
- <sup>2,3</sup> *This property is required if, and shall be present only if, the object supports COV reporting.*
- <sup>4</sup> *These properties are required if the object supports intrinsic reporting.*
- <sup>5</sup> *These properties shall be present only if the object supports intrinsic reporting.*
- <sup>3,6</sup> *If this property is present, then the Reliability property shall be present.*
- <sup>4,7</sup> *This property is required if the object supports the value source mechanism.*
- <sup>5,8</sup> *These properties are required if the object supports the value source mechanism and is commandable.*
- <sup>6,9</sup> *This property shall be present only if the object supports the value source mechanism.*
- <sup>7,10</sup> *These properties shall be present only if the object supports the value source mechanism and is commandable.*
- <sup>8,11</sup> *This property shall be writable as described in Clause 19.5.*

[Change **Clause 12.54.9**, p. 503]

### **12.54.9 Status\_Flags**

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a Lighting Output object. Two of the flags are associated with the values of other properties of this object. A more detailed status could be determined by reading the properties that are linked to these flags. The relationship between individual flags is not defined by the protocol. The four flags are:

{IN\_ALARM, FAULT, OVERRIDDEN, OUT\_OF\_SERVICE}

where:

IN_ALARM	<del>Always Logical FALSE (0).</del> <i>Logical TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).</i>
FAULT	<i>Logical TRUE (1) if the Reliability property is present and does not have a value of NO_FAULT_DETECTED, otherwise logical FALSE (0).</i>
OVERRIDDEN	<i>Logical TRUE (1) if the output has been overridden by some mechanism local to the BACnet Device, otherwise logical FALSE (0). In this context "overridden" is taken to mean that the physical output is no longer tracking changes to the Present_Value property, and the Reliability property is no longer a reflection of the physical output.</i>
OUT_OF_SERVICE	<i>Logical TRUE (1) if the Out_Of_Service property has a value of TRUE, otherwise logical FALSE (0).</i>

[Insert new **Clause 12.54.10**, p. 504]

### **12.54.10 Event\_State**

The Event\_State property, of type BACnetEventState, is included in order to provide a way to determine whether this object has an active event state associated with it (see Clause 13.2.2.1). If the object supports event reporting, then the Event\_State property shall indicate the event state of the object. If the object does not support event reporting, then the value of this property shall be NORMAL.

[Renumber **Existing Clauses 12.54.10 to 12.54.27**, p. 504]

#### ~~12.54.10~~ 12.54.11 Reliability

...

[Continue renumbering all subsequent clauses]

#### ~~12.54.27~~ 12.54.28 COV\_Increment

...

[Insert new Clauses 12.54.29 to 12.54.36, p. 504]

#### 12.54.29 Event\_Detection\_Enable

This property, of type BOOLEAN, indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services.

This property is expected to be set during system configuration and is not expected to change dynamically.

When this property is FALSE, Event\_State shall be NORMAL, and the properties Acked\_Transitions, Event\_Time\_Stamps, and Event\_Message\_Texts shall be equal to their respective initial conditions.

#### 12.54.30 Notification\_Class

This property, of type Unsigned, shall specify the instance of the Notification Class object to use for event notification-distribution.

#### 12.54.31 Event\_Enable

This property, of type BACnetEventTransitionBits, shall convey three flags that separately enable and disable the distribution of TO\_OFFNORMAL, TO\_FAULT, and TO\_NORMAL notifications (see Clause 13.2.5). A device is allowed to restrict the set of supported values for this property but shall support (T, T, T) at a minimum.

#### 12.54.32 Acked\_Transitions

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). Each flag shall have the value TRUE if no event of that type has ever occurred for the object.

#### 12.54.33 Notify\_Type

This property, of type BACnetNotifyType, shall convey whether the notifications generated by the object should be Events or Alarms. The value of the property is used as the value of the 'Notify Type' service parameter in event notifications generated by the object.

#### 12.54.34 Event\_Time\_Stamps

This read-only property, of type BACnetARRAY[3] of BACnetTimeStamp, shall convey the times of the last TO\_OFFNORMAL, TO\_FAULT, and TO\_NORMAL events (see Clause 13.2.2.1). Timestamps of type Time or Date shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

#### 12.54.35 Event\_Message\_Texts

This read-only property, of type BACnetARRAY[3] of CharacterString, shall convey the message text values of the last TO\_OFFNORMAL, TO\_FAULT, and TO\_NORMAL events (see Clause 13.2.2.1). If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

#### 12.54.36 Event\_Message\_Texts\_Config

This property, of type BACnetARRAY[3] of CharacterString, contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO\_OFFNORMAL, TO\_FAULT, and TO\_NORMAL events,

respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

[Renumber **Existing Clauses 12.54.28 to 12.54.27**, p. 504]

~~12.54.28~~ **12.54.37 Reliability\_Evaluation\_Inhibit...**

[Continue renumbering all subsequent clauses]

~~12.54.37~~ **12.54.46 Profile\_Name**

#### 135-2016br-4. Deprecate Time form of timestamps.

##### Rationale

There is no burden to track the date when the time is tracked. Timestamp values of type Time are ambiguous when used by an application where the BACnetTimeStamp data type is specified. It matters whether the timestamp intended to indicate 5 minutes ago, or one day and 5 minutes ago.

The Time form of timestamps is deprecated.

[Change **Clause 12.1.7**, p. 156]

##### 12.1.7 Unspecified Dates and Times

...

Several object types defined in this clause have properties that contain timestamp values. If no event or operation has yet occurred, then timestamp values of type BACnetDateTime shall have an unspecified datetime value, ~~timestamp values of type Time shall have an unspecified time value,~~ and timestamp values of type Unsigned shall have a value of zero. If the event or operation has occurred, then the timestamp value shall have a specific datetime value, ~~a specific time value,~~ or a value greater than zero, respectively. If a device supports the Local\_Date and Local\_Time properties, then all timestamps created by the device shall use the BACnetDateTime form. For interoperability with products claiming a protocol revision prior to 12, it is recommended that implementations accept time values with trailing unspecified octets, such as unspecified hundredths of a second.

[Change **Clause 12**, All Clauses for the **Event\_Time\_Stamps** property in every object type, p. 161]

##### 12.X.Y Event\_Time\_Stamps

This read-only property, of type BACnetARRAY[3] of BACnetTimeStamp, shall convey the times of the last TO\_OFFNORMAL, TO\_FAULT, and TO\_NORMAL events (see Clause 13.2.2.1). Timestamps of type ~~Time or Date~~ *BACnetDateTime* shall have X'FF' in each octet and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

[Change **Clause 12.31.29**, p. 360]

##### 12.31.29 Access\_Event\_Time

This property, of type BACnetTimeStamp, indicates the most recent update time of the Access\_Event property. This property shall update its value on each update of Access\_Event. Update times of type ~~Time or Date~~ *BACnetDateTime* shall have X'FF' in each octet, and Sequence Number update times shall have the value 0 if no update has yet occurred.

...

[Change **Clause 12.36.11**, p. 349]

##### 12.36.11 Update\_Time

This property, of type BACnetTimeStamp, indicates the most recent update time when the Present\_Value was updated. This property shall update its value on each update of the Present\_Value. If no update has yet occurred, update times of type ~~Time or Date~~ *BACnetDateTime* shall have X'FF' in each octet, and Sequence Number update times shall have the value 0.

[Change **Clause 20.2.18**, p. 781]

### 20.2.18 Encoding of a Choice Value

The encoding of a CHOICE value shall be the same as the encoding of a value of the chosen type. The encoding may be primitive or constructed depending on the chosen type.

Example: CHOICE of primitive data

ASN.1 = BACnetTimeStamp  
Value = ~~5:35:45.17 P.M.~~ = 17:35:45.17 15  
Context Tag = 0 1 (Choice for ~~time~~ *sequence number*' in BACnetTimeStamp)  
Encoded Tag = X'0C 19'  
Encoded Data = X'~~41232D44~~ 0F'

...

[Change Clause 21, BACnetTimeStamp production, p. 869]

```
BACnetTimeStamp ::= CHOICE {  
    time [0] Time, -- this form was deprecated in version 1 revision X  
    sequenceNumber [1] Unsigned (0..65535),  
    dateTime [2] BACnetDateTime  
}
```

### 135-2016br-5. Clarify the Multi-state object types when Number\_Of\_States shrinks.

#### Rationale

The Number\_Of\_States property defines the number of states the Present\_Value may have, but that is not required to be maintained on a dynamic basis. The Priority\_Array entries, Relinquish\_Default, Alarm\_Values and Fault\_Values, properties are similarly not required to be updated when the Number\_Of\_States property changes to a value lower than their current value(s).

In these same paragraphs is language that has been differently interpreted regarding whether Present\_Value is or is not required to be capable of taking on all values between 1 and Number\_Of\_States.

The behavior on when the number of states shrinks is clarified. Also, language is clarified on the support of values between 1 and Number\_Of\_States.

[Insert in **Clause 12.1.8**, p. 157]

#### 12.1.8 Reliability

...

*MULTI\_STATE\_OUT\_OF\_RANGE*

*The present value of the Multi-state object is greater than the value of Number\_of\_States, for example, when the number of states becomes less than the Present\_Value's current value.*

*UNRELIABLE\_OTHER*

The controller has detected that the present value is unreliable, but none of the other conditions describe the nature of the problem. A generic fault other than those listed above has been detected, e.g., a Binary Input is not cycling as expected.

[Change **Clause 12.18.11**, p. 259]

#### 12.18.11 Number\_Of\_States

This property, of type Unsigned, defines the *upper bound of the* number of states that the Present\_Value may have. The Number\_Of\_States property shall always have a value greater than zero. If the value of this property is changed, the size of the State\_Text array, if present, shall also be changed to the same value. *If the Number\_of\_States property value becomes less than the Present\_Value, the object shall have a Reliability of MULTI\_STATE\_OUT\_OF\_RANGE as long as this situation remains, unless the object is out of service. It is a local matter whether Present\_Value is modified when the Number\_of\_States property value becomes less than its current value.*

[Change **Clause 12.19.11**, p. 264]

#### 12.19.11 Number\_Of\_States

This property, of type Unsigned, defines the *upper bound of the* number of states *that* the Present\_Value may have. The Number\_Of\_States property shall always have a value greater than zero. If the value of this property is changed, the size of the State\_Text array, if present, shall also be changed to the same value. *If the Number\_of\_States property value becomes less than the Present\_Value, the object shall have a Reliability of MULTI\_STATE\_OUT\_OF\_RANGE as long as this situation remains, unless the object is out of service. It is a local matter whether Priority\_Array, Relinquish\_Default, Present\_Value, and Feedback\_Value properties are modified when the Number\_of\_States property value becomes less than their current values. If any of those properties other than Present\_Value are out of range, the value of the Reliability property shall remain CONFIGURATION\_ERROR, unless the object is out of service.*

[Change **Clause 12.20.10**, p. 270]

#### 12.20.10 Number\_Of\_States

This property, of type Unsigned, defines the *upper bound of the* number of states that the Present\_Value may have. The Number\_Of\_States property shall always have a value greater than zero. If the value of this property is changed, the size of the State\_Text array, if present, shall also be changed to the same value. *If the Number\_of\_States property value becomes less than the value of the Present\_Value, the object shall have a Reliability of MULTI\_STATE\_OUT\_OF\_RANGE as long as this situation remains, unless the object is out of service. It is a local matter whether Priority\_Array, Relinquish\_Default, Present\_Value, Alarm\_Values, and Fault\_Values properties, if present, are modified when the Number\_of\_States property value becomes less than their current values. If any of those properties other than Present\_Value are out of range, the value of the Reliability property shall remain CONFIGURATION\_ERROR, unless the object is out of service.*

[Change **Clause 21, BACnetReliability**, p. 865]

**BACnetReliability** ::= ENUMERATED {

```
...
referenced-object-fault      (24),
multi-state-out-of-range     (25),
...
```

```
...
}
```

```
-- Enumerated values 0-63 are reserved for definition by ASHRAE. Enumerated values
-- 64-65535 may be used by others subject to the procedures and constraints described
-- in Clause 23.
```

**135-2016br-6. Fix the language for event type and message text parameters of event notifications.**

**Rationale**

In Table 13-3, the 'Event Type' parameter in the Acknowledgment Transition column is contradicting the structure of the ConfirmedEventNotification request in Clause 13.8.1 and the structure of the UnconfirmedEventNotification request in Clause 13.9.1

The definition of the 'Event Type' parameter in Clauses 13.8.1.1.7 and 13.9.1.1.7 does not mention the 'Event Type' for transitions to the FAULT event state and from the FAULT event state.

Also for the 'Message Texts' parameter of these services, clarify their requirement to derive the value from the value configured in the corresponding transition in the Event\_Message\_Texts\_Config property, if that property exists, and to place the value into the corresponding Event\_Message\_Texts, if that property exists.

[Change **Clause 13.2.5.2**, p. 599]

**13.2.5.2 Service Parameters of Event Notification Service Requests**

...

**Table 13-3.** Event Notification Service Parameter Values

Service Parameter	Event State Transition (all transitions)	Acknowledgment Transition
...	...	...
Event Type	When 'To State' or 'From State' is FAULT, set to CHANGE_OF_RELIABILITY,  Otherwise the value associated with the event-initiating object's <i>configured</i> event algorithm.	<del>Not present</del> When 'To State' is FAULT, set to CHANGE_OF_RELIABILITY.  When 'To State' is NORMAL, and the device can determine reporting acknowledgement of a transition from FAULT, set to CHANGE_OF_RELIABILITY.  Otherwise the value associated with the event-initiating object's <i>configured</i> event algorithm.
Message Text	Optional  The value is <i>derived from the value in the Event_Message_Texts_Config property configured in the entry corresponding to the transition, if the property exists. Otherwise the value is a local matter. The Message Text transmitted</i> is reflected into the Event_Message_Texts array, if the property exists.	Optional  The value is a local matter.
...	...	...

[Change **Clause 13.8.1.1.7 and 13.8.1.1.8**, p. 644]

#### **13.8.1.1.7 Event Type**

This parameter, of type BACnetEventType, shall specify the type of event that has occurred.

*If 'Notify Type' is ALARM or EVENT when the 'To State' parameter is FAULT, then this parameter shall have a value of CHANGE\_OF\_RELIABILITY. The Event Type CHANGE\_OF\_RELIABILITY shall be used for reporting a transition from FAULT. Otherwise, this parameter shall have the value associated with the event-initiating object's configured event algorithm.*

*If 'Notify Type' is ACK\_NOTIFICATION when 'To State' is FAULT, the Event Type shall be CHANGE\_OF\_RELIABILITY. When 'To State' is NORMAL, and the device can determine reporting acknowledgement of a transition from FAULT, the Event Type shall be CHANGE\_OF\_RELIABILITY. Otherwise, the Event Type shall be the value associated with the event-initiating object's configured event algorithm.*

#### **13.8.1.1.8 Message Text**

This optional parameter, of type CharacterString, shall convey a string of printable characters. This parameter may be used to convey a message to be logged or displayed, which pertains to the occurrence of the event.

*If 'Notify Type' is ALARM or EVENT, the content of the message is a local matter. text shall be derived from the value in the Event\_Message\_Texts\_Config entry corresponding to the transition, if the property exists. Otherwise, the value is a local matter. If the optional property Event\_Message\_Texts is present in the event-initiating generating object, the text conveyed in this Message Text parameter shall be stored in the respective field of the Event\_Message\_Texts array.*

*If 'Notify Type' is ACK\_NOTIFICATION, the value is a local matter.*

[Change **Clause 13.9.1.1.7 and 13.9.1.1.8** p. 647]

#### **13.9.1.1.7 Event Type**

This parameter, of type BACnetEventType, shall specify the type of event that has occurred.

*If 'Notify Type' is ALARM or EVENT when the 'To State' parameter is FAULT, then this parameter shall have a value of CHANGE\_OF\_RELIABILITY. The Event Type CHANGE\_OF\_RELIABILITY shall be used for reporting a transition from FAULT. Otherwise, this parameter shall have the value associated with the event-initiating object's configured event algorithm.*

*If 'Notify Type' is ACK\_NOTIFICATION when 'To State' is FAULT, the Event Type shall be CHANGE\_OF\_RELIABILITY. When 'To State' is NORMAL, and the device can determine reporting acknowledgement of a transition from FAULT, the Event Type shall be CHANGE\_OF\_RELIABILITY. Otherwise, the Event Type shall be the value associated with the event-initiating object's configured event algorithm.*

#### **13.9.1.1.8 Message Text**

This optional parameter, of type CharacterString, shall convey a string of printable characters. This parameter may be used to convey a message to be logged or displayed, which pertains to the occurrence of the event.

*If 'Notify Type' is ALARM or EVENT, the content of the message is a local matter. text shall be derived from the value in the Event\_Message\_Texts\_Config property entry corresponding to the transition, if the property exists. Otherwise, the value is a local matter. If the optional property Event\_Message\_Texts is present in the event-initiating generating object, the text conveyed in this Message Text parameter shall be stored in the respective field of the Event\_Message\_Texts array.*

*If 'Notify Type' is ACK\_NOTIFICATION, the value is a local matter.*

**135-2016br-7. Clarify the object instance 4194303.**

**Rationale**

The object instance 4194303 is not only used to indicate an uninitialized object identifier. Several other uses of this instance are specified in the standard.

The language for the instance 4194303 is changed to reflect these uses.

[Change **Clause 12.1.1**, p. 153]

**12.1.1 Identification of Objects**

All objects are referenced by their Object\_Identifier property and every object shall have an Object\_Identifier property. The Object\_Identifier is composed of two parts: an object type and an object instance (see Clause 20.2.14). Each object within a single BACnet device shall have a unique value for the Object\_Identifier property. No object shall have an Object\_Identifier with an instance number of 4194303. Object properties that contain values whose datatype is BACnetObjectIdentifier may use 4194303 as the instance number to indicate that the property is ~~not initialized~~, *uninitialized, disabled, or unused, except where noted in individual clauses.*

### 135-2016br-8. Wildcard instance for Network Port objects in ReadPropertyMultiple requests

#### Rationale

When the Network Port object was added to the standard, the ReadProperty service was extended such that when the 'Object Identifier' parameter contains the value (NETWORK\_PORT, 4194303), the responding BACnet-user shall treat the 'Object Identifier' as if it correctly matched the local Network Port object. This was specified only in the ReadProperty service, despite that a very similar treatment for Device object was specified in both ReadProperty and ReadPropertyMultiple services.

The ReadPropertyMultiple service is extended to also support the Network Port wildcard instance treatment.

[Change **Clause 15.7.2**, p. 689]

#### 15.7.2 Service Procedure

...

When the object-type in the Object Identifier portion of the Read Access Specification parameter contains the value DEVICE and the instance of that 'Object Identifier' parameter contains the value 4194303, the responding BACnet-user shall treat the Object Identifier as if it correctly matched the local Device object. This allows the device instance of a device that does not generate I-Am messages to be determined.

*When the object-type in the Object Identifier portion of the Read Access Specification contains the value NETWORK\_PORT and the instance of that 'Object Identifier' parameter contains the value 4194303, the responding BACnet-user shall treat the Object Identifier as if it correctly matched the local Network Port object representing the network port through which the request was received. This allows the network port instance of the network port that was used to receive the request to be determined.*

**135-2016br-9. Clarify the timestamp of trend log and trend log multiple log records.**

**Rationale**

In the trend log objects, the language on the time stamp of log records is unclear and does not address the different logging types.

The time stamp in trend log and trend log multiple records is clarified in relation to the logging type.

[Change in **Clause 12.25.14**, p. 302]

**12.25.14 Log\_Buffer**

This property, of type BACnetLIST of BACnetLogRecord, is a list of up to Buffer\_Size timestamped log records of datatype BACnetLogRecord, each of which conveys a recorded data value, an error related to data-collection, or status changes in the Trend Log object. Each log record has data fields as follows:

Timestamp	<del>The local date and time when the log record was collected.</del> <i>When Logging_Type is POLLED or TRIGGERED the timestamp shall be the date and time the logging object attempts to acquire the record. When Logging_Type is COV, the timestamp shall be the date and time included within the notification, if supplied otherwise, the date and time when the notification was received.</i>
...	

[Change **Clause 12.30.19**, p. 343]

**12.30.19 Log\_Buffer**

This property, of type BACnetLIST of BACnetLogMultipleRecord, is a list of log records where each log record conveys either a set of recorded data values or errors related to data-collection, a status change in the Trend Log Multiple object, or an indication that the time and/or date was changed in the device hosting the Trend Log Multiple object. Each log record has data fields as follows:

Timestamp	The local date and time when the <i>logging object attempts to acquire the record</i> <del>was stored.</del>
...	

[Add a new entry to History of Revisions, p. 1364]

## HISTORY OF REVISIONS

...	...	...
1	21	<p><b>Addendum <i>br</i> to ANSI/ASHRAE Standard 135-2016</b> Approved by ASHRAE and by the American National Standards Institute on August 26, 2019.</p> <ol style="list-style-type: none"><li>1. Add new engineering units.</li><li>2. Add mandate to accept writes of NULL to non-commandable properties.</li><li>3. Add intrinsic fault reporting to Lighting Output object type.</li><li>4. Deprecate Time form of timestamps.</li><li>5. Clarify the Multi-state object types when Number_Of_States shrinks.</li><li>6. Fix the language for event type and message text parameters of event notifications.</li><li>7. Clarify the object instance 4194303.</li><li>8. Wildcard instance for Network Port objects in ReadPropertyMultiple requests.</li><li>9. Clarify the timestamp of trend log and trend log multiple log records.</li></ol>

## **POLICY STATEMENT DEFINING ASHRAE'S CONCERN FOR THE ENVIRONMENTAL IMPACT OF ITS ACTIVITIES**

ASHRAE is concerned with the impact of its members' activities on both the indoor and outdoor environment. ASHRAE's members will strive to minimize any possible deleterious effect on the indoor and outdoor environment of the systems and components in their responsibility while maximizing the beneficial effects these systems provide, consistent with accepted Standards and the practical state of the art.

ASHRAE's short-range goal is to ensure that the systems and components within its scope do not impact the indoor and outdoor environment to a greater extent than specified by the Standards and Guidelines as established by itself and other responsible bodies.

As an ongoing goal, ASHRAE will, through its Standards Committee and extensive Technical Committee structure, continue to generate up-to-date Standards and Guidelines where appropriate and adopt, recommend, and promote those new and revised Standards developed by other responsible organizations.

Through its *Handbook*, appropriate chapters will contain up-to-date Standards and design considerations as the material is systematically revised.

ASHRAE will take the lead with respect to dissemination of environmental information of its primary interest and will seek out and disseminate information from other responsible organizations that is pertinent, as guides to updating Standards and Guidelines.

The effects of the design and selection of equipment and systems will be considered within the scope of the system's intended use and expected misuse. The disposal of hazardous materials, if any, will also be considered.

ASHRAE's primary concern for environmental impact will be at the site where equipment within ASHRAE's scope operates. However, energy source selection and the possible environmental impact due to the energy source and energy transportation will be considered where possible. Recommendations concerning energy source selection should be made by its members.

## About ASHRAE

Founded in 1894, ASHRAE is a global professional society committed to serve humanity by advancing the arts and sciences of heating, ventilation, air conditioning, refrigeration, and their allied fields.

As an industry leader in research, standards writing, publishing, certification, and continuing education, ASHRAE and its members are dedicated to promoting a healthy and sustainable built environment for all, through strategic partnerships with organizations in the HVAC&R community and across related industries.

To stay current with this and other ASHRAE Standards and Guidelines, visit [www.ashrae.org/standards](http://www.ashrae.org/standards), and connect on LinkedIn, Facebook, Twitter, and YouTube.

## Visit the ASHRAE Bookstore

ASHRAE offers its Standards and Guidelines in print, as immediately downloadable PDFs, and via ASHRAE Digital Collections, which provides online access with automatic updates as well as historical versions of publications. Selected Standards and Guidelines are also offered in redline versions that indicate the changes made between the active Standard or Guideline and its previous version. For more information, visit the Standards and Guidelines section of the ASHRAE Bookstore at [www.ashrae.org/bookstore](http://www.ashrae.org/bookstore).

### IMPORTANT NOTICES ABOUT THIS STANDARD

**To ensure that you have all of the approved addenda, errata, and interpretations for this Standard, visit [www.ashrae.org/standards](http://www.ashrae.org/standards) to download them free of charge.**

**Addenda, errata, and interpretations for ASHRAE Standards and Guidelines are no longer distributed with copies of the Standards and Guidelines. ASHRAE provides these addenda, errata, and interpretations only in electronic form to promote more sustainable use of resources.**