ADDENDA

ANSI/ASHRAE Addendum bd to
ANSI/ASHRAE Standard 135-2016

A Data Communication Protocol for Building Automation and Control Networks

Approved by ASHRAE on June 15, 2018, and by the American National Standards Institute on June 15, 2018.

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. The change submittal form, instructions, and deadlines may be obtained in electronic form from the ASHRAE® website (www.ashrae.org) or in paper form from the Senior Manager of Standards.

The latest edition of an ASHRAE Standard may be purchased on the ASHRAE website (www.ashrae.org) or from ASHRAE Customer Service, 1791 Tullie Circle, NE, Atlanta, GA 30329-2305. E-mail: 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.

 ISSN 1041-2336

Approved American National Standard
ANSI
www.ansi.org

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

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

[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 changes to ANSI/ASHRAE Standard 135-2016. 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 changes are summarized below.

135-2016*bd*-1 Add a Staging Object Type, p. 2

In the following document, language to be added to existing clauses of ANSI/ASHRAE Standard 135-2016 is indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are added, plain type is used throughout.

The use of placeholders like X, Y, Z, X1, X2, etc., should not be interpreted as literal values of the final standard. These placeholders will be assigned actual numbers/letters only with incorporation of this addendum into the standard for republication.

135-2016*bd*-1 Add a Staging Object Type

Rationale

The Staging object type provides a way for BACnet devices to map analog values onto multiple Binary Value, Binary Output, or Binary Lighting Output objects.

A common use case is in lighting applications, where a level, identified by a numeric value, sets the appropriate values of multiple binary outputs (on or off).

Support of this new object type is excluded from all data sharing BIBBs for life safety and access control.

[Insert new **Clause 12.X**, p. 584]

12.X Staging Object Type

The Staging object type defines a standardized object whose properties represent the externally visible characteristics of a staged value. A "Staging" maps a numeric value onto multiple discrete ranges that define individual “stages” ($\mathbf{N_{stages}}$). Each Staging object is associated with a collection of references to binary valued objects ($\mathbf{N_{references}}$). Each Staging object may therefore control Binary Output, Binary Value, or Binary Lighting Output objects. Every stage specifies an arbitrary combination of ACTIVE/INACTIVE values to be written to these referenced objects. Stages are defined by a limit, a deadband, and the collection of values for the referenced objects.

Figure 12-X shows a typical Staging object application with four stages ($\mathbf{N_{stages}}$ =4) and two referenced binary objects ($\mathbf{N_{references}}$ = 2).

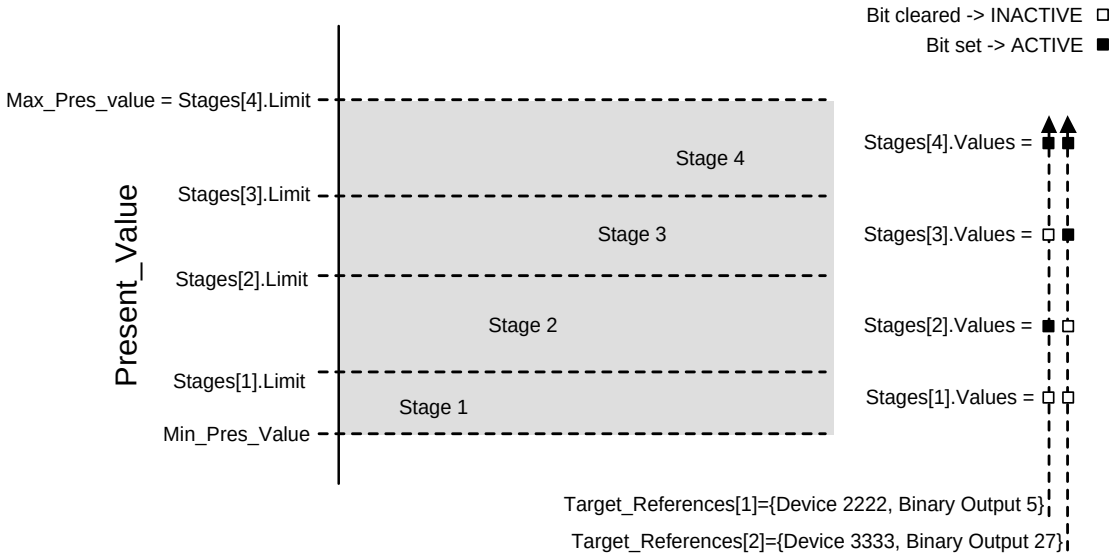

Figure 12-X. Typical Staging Application ($\mathbf{N_{stages}}$ =4, $\mathbf{N_{references}}$ = 2)

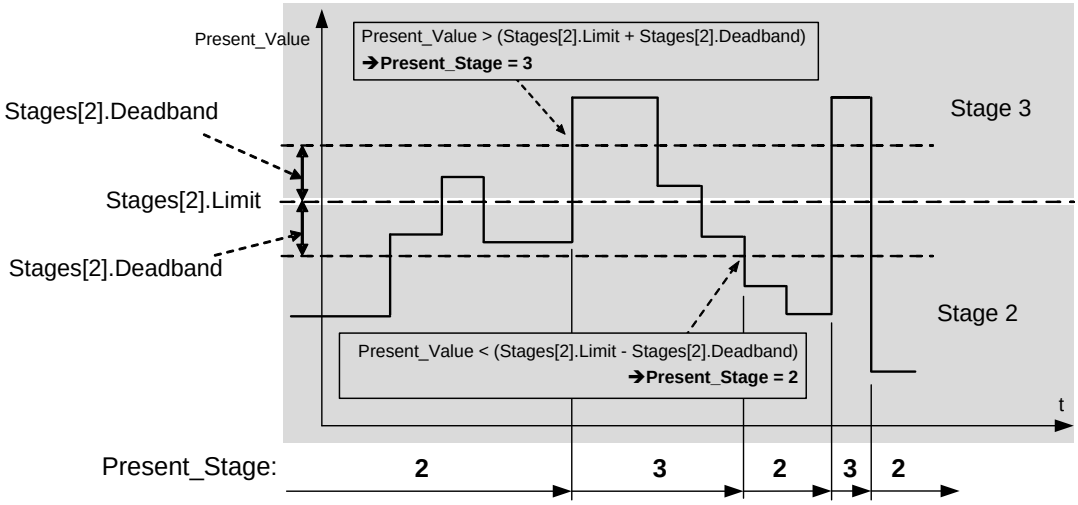

Figure 12-X2. Stage Limits incorporate hysteresis through the use of a Deadband around each Limit

Stages are defined by limits with a symmetrical deadband. A deadband greater than zero is used to prevent unwanted oscillation when the Present_Value is close to a limit. As the Present_Value increases, if it rises above the limit for a stage plus the deadband for that stage, the Present_Stage transitions to that stage+1. Similarly, as the Present_Value decreases, it must fall below the limit for a stage minus the deadband for that stage before Present_Stage transitions to that stage. The deadband is allowed to be zero (0.0).

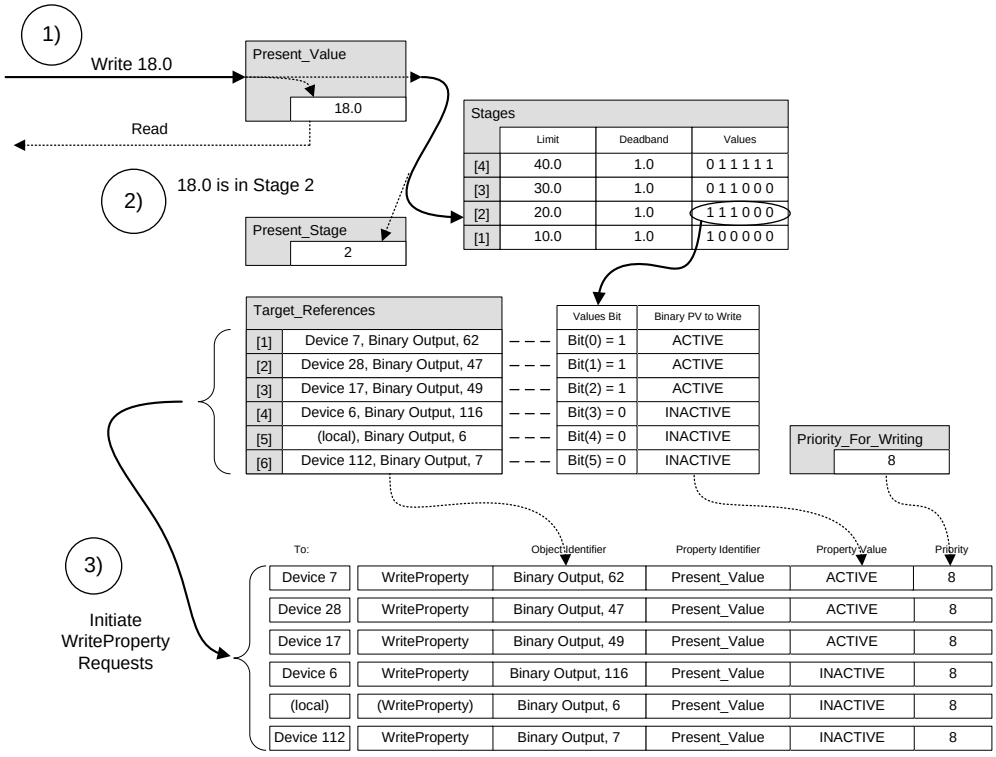

Figure 12-X3. Pipeline of operations when Present_Value is written

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

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

Table 12-X. Properties of the Staging Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	REAL	W
Present_Stage	Unsigned	R
Stages	BACnetARRAY[N] of BACnetStageLimitValue	R[1]
Stage_Names	BACnetARRAY[N] of CharacterString	O[1]
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	R
Out_Of_Service	BOOLEAN	R
Description	CharacterString	O
Units	BACnetEngineeringUnits	R
Target_References	BACnetARRAY[N] of BACnetDeviceObjectReference	R[2]
Priority_For_Writing	Unsigned(1..16)	R
Default_Present_Value	REAL	O
Min_Pres_Value	REAL	R
Max_Pres_Value	REAL	R
COV_Increment	REAL	O[3]
Notification_Class	Unsigned	O[4,5]
Event_Enable	BACnetEventTransitionBits	O[4,5]
Acked_Transitions	BACnetEventTransitionBits	O[4,5]
Notify_Type	BACnetNotifyType	O[4,5]
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	O[4,5]
Event_Message_Texts	BACnetARRAY[3] of CharacterString	O[5,6]
Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	O[5]
Event_Detection_Enable	BOOLEAN	O[4,5]
Reliability_Evaluation_Inhibit	BOOLEAN	O
Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R
Value_Source	BACnetValueSource	O[7,8,9]
Tags	BACnetARRAY[N] of BACnetNameValue	O
Profile_Location	CharacterString	O
Profile_Name	CharacterString	O

[1] The array size of this property is $\mathbf{N_{stages}}$.
[2] The array size of this property is $\mathbf{N_{references}}$.
[3] This property is required if the object supports COV reporting.
[4] These properties are required if the object supports intrinsic reporting.
[5] These properties shall be present only if the object supports intrinsic reporting.
[6] This property, if present, is required to be read-only.
[7] This property is required if the object supports the value source mechanism.
[8] This property shall be present only if the object supports the value source mechanism.
[9] This property shall be writable as described in Clause 19.5.

12.X.1 Object_Identifier

This property, of type BACnetObjectIdentifier, is a numeric code that is used to identify the object. It shall be unique within the BACnet device that maintains it.

12.X.2 Object_Name

This property, of type CharacterString, shall represent a name for the object that is unique within the BACnet device that maintains it. The minimum length of the string shall be one character. The set of characters used in the Object_Name shall be restricted to printable characters.

12.X.3 Object_Type

This property, of type BACnetObjectType, indicates membership in a particular object type class. The value of this property shall be STAGING.

12.X.4 Present_Value

This property, of type REAL, indicates the current value, in engineering units, of the Staging object. If Present_Value is written with a value less than Min_Pres_Value, then it shall be clamped to Min_Pres_Value. If Present_Value is written with a value greater than Max_Pres_Value, then it shall be clamped to Max_Pres_Value.

Whenever Present_Value is changed, the new value shall be compared with the 'Limit' values for the entire Stages array using the following algorithm:

```
ops = current Present_Stage
npv = new Present_Value

//check if value should remain in the current stage
If (ops != 0) then
      upperBound = Stages[ops].Limit + Stages[ops].Deadband
      if (ops > 1) then
            lowerBound = Stages[ops-1].Limit - Stages[ops-1].Deadband
      else
            lowerBound = Min_Pres_Value
      endif
      if (npv <= upperBound AND npv >= lowerBound) then
            Present_Value = npv
            exit algorithm    //no change to current stage, stop algorithm
      endif
endif

// calculate the new stage
Present_Stage = Nstages
for (i=1 to Nstages-1, step +1)
      if npv <= (Stages[i].Limit) then
            Present_Stage = i
            exit for          //found the correct stage, stop iteration
      endif
next i

Present_Value = npv
```

Figure 12-X4. Pseudocode Algorithm for Evaluating Present_Value and Present_Stage

12.X.4.1 Writing to Referenced Objects

Changes to Present_Stage shall cause a write of ACTIVE or INACTIVE to the corresponding object's Present_Value for each Target_References array element. For each bit (Index = 0 to $\mathbf{N}_{\mathbf{references}}$ -1) in the Stages[Present_Stage].Values bitstring, if the bit is set (1), an ACTIVE value shall be written, or if clear (0), then an INACTIVE value shall be written to the Target_References[Index +1] object's Present_Value.

Writes to Present_Value that subsequently trigger writing to referenced objects due to reevaluation of Present_Stage, are not expected to wait until the reference writes occur before returning a Result(+) or Result(-) for the write to Present_Value. Subsequently if any write to a referenced object fails, Reliability shall be changed to COMMUNICATION_FAILURE. The COMMUNICATION_FAILURE shall remain in effect until all reference writes have been completed successfully. How a particular implementation handles other failures during writing to referenced objects shall be a local matter except that Reliability shall indicate a value other than NO_FAULT_DETECTED.

The order of evaluation of references and any referenced object write delay shall be a local matter.

12.X.5 Present_Stage

This property, of type Unsigned, shall indicate the array index (1 to $\mathbf{N}_{\mathbf{stages}}$) that corresponds to the current active stage or 0 meaning that the Present_Stage has not yet been initialized. Upon device restart, or when the Stages property is written to any of its elements, or the size of the Stages array changes, Present_Stage shall be set to 0 temporarily and then Present_Value shall be reevaluated as described in Clause 12.X.4.1.

Attempts to read Present_Stage when it is internally set to 0 shall return a Result(-) with an 'Error Class' of PROPERTY and an 'Error Code' of VALUE_NOT_INITIALIZED.

12.X.6 Stages

This property, of type BACnetARRAY[N] of BACnetStageLimitValue, is an array representing the stages by limit, desired present values for the objects referenced by Target_References, and deadband. The size of the array is $\mathbf{N}_{\mathbf{stages}}$, where $\mathbf{N}_{\mathbf{stages}}$ shall be greater than 1. BACnetStageLimitValue is a tuple consisting of the following fields:

Limit	REAL
Values	BIT STRING
Deadband	REAL

The 'Limit' values for all elements shall be strictly ascending, such that:

```
for Index=1 to Nstages-1
{
    lowerbound = (Stages[Index].Limit + Stages[Index].Deadband)
    upperbound = (Stages[Index+1].Limit - Stages[Index+1].Deadband)
    lowerbound <= upperbound
}
```

If any of the stages do not meet this criterion, then the Reliability property shall have a value of CONFIGURATION_ERROR.

The bits in 'Values' correspond to references in the Target_References array, such that bit (Index = 0 to $\mathbf{N}_{\mathbf{references}}$ - 1) corresponds to Target_References[Index + 1], etc. The length of the 'Values' bitstring shall be $\mathbf{N}_{\mathbf{references}}$ bits.

The 'Deadband' shall be zero or positive. A negative value for 'Deadband' shall cause the value of Reliability to be CONFIGURATION_ERROR.

If the size of the Stages array is increased, then the new array elements, for which no initial value is provided, shall be initialized to contain 'Limit' = 0.0, 'Deadband' = 0.0, and 'Values' = {0…0}, and the value of Reliability shall be set to CONFIGURATION_ERROR.

If the size of the Stages array is less than 2, then the Reliability property shall have a value of CONFIGURATION_ERROR.

If Reliability has the value CONFIGURATION_ERROR, then Present_Value shall be set to Min_Pres_Value and Present_Stage to 1.

If Stages[$\mathbf{N_{stages}}$].Limit becomes smaller than Present_Value, then Present_Value shall be set to Stages[$\mathbf{N_{stages}}$].Limit.

If the Stages property is written, the value of Present_Stage shall be reevaluated and corresponding writes of new values to Target_References' objects shall be triggered.

If the size of this array is changed, the size of the Stage_Names array shall also be changed to the same size.

12.X.7 Stage_Names

This property, of type BACnetARRAY[N] of CharacterString, is an array representing a name for each stage. The number of array elements in Stage_Names shall be the same as the number of array elements in the Stages property. Stage_Names[1] shall correspond to the name for Present_Stage=1. Stage_Names[2] shall correspond to the name for Present_Stage=2, etc.

If the size of this array is changed, the size of the Stages array shall also be changed to the same size. If the size of Stage_Names is increased, then it shall be a local matter what the uninitialized array elements contain.

12.X.8 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a Staging object. Three 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	Logical FALSE (0) if the Event_State property has a value of NORMAL, otherwise logical TRUE (1).
FAULT	Logical TRUE (1) if the Reliability property does not have a value of NO_FAULT_DETECTED, otherwise logical FALSE (0).
OVERRIDDEN	Always logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property has a value of TRUE, otherwise logical FALSE(0).

If the object supports event reporting, then this property shall be the pStatusFlags parameter for the object's event algorithm. See Clause 13.3 for event algorithm parameter descriptions.

12.X.9 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.

12.X.10 Reliability

The Reliability property, of type BACnetReliability, provides an indication that the properties of the Staging object are in a consistent state and that Target_References are being reliably written. See Clauses 12.X.4.1, 12.X.6, and 12.X.17.

Table 12-X2 summarizes scenarios when Reliability has a value other than NO_FAULT_DETECTED.

Table 12-X2. Reliability Scenarios

Scenario	Reliability Value	See Clause
Communication error when writing	COMMUNICATION_FAILURE	12.X.4.1
Non-communication-related failure when writing	Local Matter	12.X.4.1
'Limit' values out of order	CONFIGURATION_ERROR	12.X.6
'Deadband' is negative	CONFIGURATION_ERROR	12.X.6
Stages array size is increased	CONFIGURATION_ERROR	12.X.6
Min_Pres_Value is >= (Stages[1].Limit - Stages[1].Deadband)	CONFIGURATION_ERROR	12.X.17

12.X.11 Out_Of_Service

The Out_Of_Service property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present_Value property is controllable by software local to the BACnet device.

When Out_Of_Service is TRUE:

a) changes to the Present_Value property are decoupled from the Target_References. This means that the objects referenced by Target_References shall not be updated;
b) the Present_Value property and the Reliability property, if capable of taking on values other than NO_FAULT_DETECTED, shall be writable to allow simulating specific conditions or for testing purposes;
c) other functions that depend on the state of the Present_Value or Reliability properties shall respond to changes made to these properties, as if those changes had occurred while the object was in service;

Restrictions on writing to the Present_Value property by software local to the BACnet device do not apply to local human-machine interfaces.

When Out_Of_Service becomes FALSE, it shall trigger updating of referenced objects as described in Clause 12.X.4.1.

12.X.12 Description

This property, of type CharacterString, is a string of printable characters whose content is not restricted.

12.X.13 Units

This property, of type BACnetEngineeringUnits, indicates the measurement units of this object. See the BACnetEngineeringUnits ASN.1 production in Clause 21 for a list of engineering units defined by this standard.

12.X.14 Target_References

This property, of type BACnetARRAY[N] of BACnetDeviceObjectReference, is an array representing references to Binary Output, Binary Value, or Binary Lighting Output objects. The size of the array is $\mathbf{N}_{\mathbf{references}}$. The Target_References array elements [Index = 1 to $\mathbf{N}_{\mathbf{references}}$] shall correspond to the bit(Index - 1) of all 'Values' bitstrings of the Stages property.

If the property is restricted to referencing objects within the containing device, an attempt to write to a reference to an object outside the containing device into this property shall cause a Result(-) to be returned with an error class of PROPERTY and an error code of OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED. Uninitialized Target_References array elements shall be given the instance number 4194303. A Target_References array element whose instance number is equal to 4194303 shall be considered uninitialized and shall be ignored in all operations that may use that reference.

If the size of this array is changed, the size of the 'Values' bitstrings of the Stages property shall also be changed to the same size. If the bitstring length is increased, added bits shall be cleared (0).

12.X.15 Priority_For_Writing

This property defines the priority at which the referenced properties are commanded. It corresponds to the 'Priority' parameter of the WriteProperty service. It is an unsigned integer in the range 1-16, with 1 being considered the highest priority and 16 the lowest. See Clause 19.2.

12.X.16 Default_Present_Value

This optional property, of type REAL, defines the value to be used for Present_Value upon device restart. Upon restart Default_Present_Value shall be copied to Present_Value and cause the reevaluation of Present_Value as described in Clause 12.X.4.

12.X.17 Min_Pres_Value

This property, of type REAL, represents the minimum value for Present_Value. Min_Pres_Value shall always be strictly less than the quantity (Stages[1].Limit - Stages[1].Deadband).

If a change to Min_Pres_Value, Stages[1].Limit, or Stages[1].Deadband violates this rule, then the Reliability property shall have a value of CONFIGURATION_ERROR. If a change to Min_Pres_Value causes it to be greater than Present_Value, then Present_Value shall be written with Min_Pres_Value.

12.X.18 Max_Pres_Value

This read-only property, of type REAL, represents the maximum value for Present_Value. This value shall be defined as the 'Limit' of the last entry of the Stages property array (i.e., Stages[$\mathbf{N}_{\mathbf{stages}}$].Limit).

If the size of the Stages array is zero, then Max_Pres_Value shall be equal to Min_Pres_Value.

12.X.19 COV_Increment

This property, of type REAL, shall specify the minimum change in Present_Value that will cause a COV notification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

12.X.20 Notification_Class

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

12.X.21 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.X.22 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.X.23 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.X.24 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.X.25 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.X.26 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.

12.X.27 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.X.28 Reliability_Evaluation_Inhibit

This property, of type BOOLEAN, indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation.

When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property.

12.X.29 Property_List

This read-only property is a BACnetARRAY of property identifiers, one 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.

12.X.30 Value_Source

This property, of type BACnetValueSource, indicates the source of the value of the Present_Value. The Value_Source property and its use in the value source mechanism are described in Clause 19.5.

12.X.31 Tags

This property, of type BACnetARRAY of BACnetNameValue, is a collection of tags for the object. See Clause Y.1.4 for restrictions on the string values used for the names of these tags and for a description of tagging and the mechanism by which tags are defined.

Each entry in the array is a BACnetNameValue construct which consists of the tag name and an optional value. If the tag is defined to be a "semantic tag", then it has no value, and the "value" field of the BACnetNameValue shall be absent.

While some tags may be known in advance when a device is manufactured, it is recommended that implementations consider that this kind of information might not be known until a device is deployed and to provide a means of configuration or writability of this property.

12.X.32 Profile_Location

This property, of type CharacterString, is the URI of the location of an xdd file (See Clause X.2) containing the definition of the CSML type specified by the Profile_Name property and possible other information (See Annex X). The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. See Clause Q.8 for the definition of the "bacnet" URI scheme.

If a Profile_Location value is not provided for a particular object, then the client shall use the Profile_Location of the Device object, if provided, to find the definition of the Profile_Name.

12.X.33 Profile_Name

This property, of type CharacterString, is the name of an object profile to which this object conforms. To ensure uniqueness, a profile name shall begin with a vendor identifier code (see Clause 23) in base-10 integer format, followed by a dash. All subsequent characters are administered by the organization registered with that vendor identifier code. The vendor identifier

code that prefixes the profile name shall indicate the organization that publishes and maintains the profile. This vendor identifier need not have any relationship to the vendor identifier of the device within which the object resides.

A profile defines a set of additional properties, behavior, and/or requirements for this object beyond those specified here. This standard defines only the format of the names of profiles. If the Profile_Location property of this object or the Device object is present and nonempty, then the value of this property shall be the name of a CSML type defined in an xdd file referred to by the Profile_Location property.

[Change **Table 13-1**, p. 588]

Table 13-1. Standardized Objects That May Support COV Reporting

Object Type	Criteria	Properties Reported
…		
Staging	*If Present_Value changes by COV_Increment* *or* *Status_Flags changes at all* *or* *Present_Stage changes at all*	*Present_Value, Status_Flags, Present_Stage*

[Change **Table 13-5**, p. 600]

Table 13-5. Properties Reported in CHANGE_OF_RELIABILITY Notifications

Object Type	Properties
...	...
Staging	*Present_Value* *Present_Stage*
Timer	...

[1] ...

[Change **Clause 21**, **BACnetObjectType** and **BACnetObjectTypesSupported** productions, p. 840]

BACnetObjectType ::= ENUMERATED { -- see below for numerical order

```
        ...
        schedule                    (17)
        staging                     (60),
        structured-view             (29),
        ...
-- -numerical order reference
        ...
        -- see lift                 (59),
        -- see staging              (60),
        ...
}
```

BACnetObjectTypesSupported ::= BIT STRING {

```
        ...
        lift                        (59)
        staging                     (60)
}
```

[Change **Clause 21**, **BACnetPropertyIdentifier** production, p. 845]

BACnetPropertyIdentifier ::= ENUMERATED { -- see below for numerical order

```
        ...
        default-fade-time               (374),
        default-present-value           (492),
        default-ramp-rate               (375),
        ...
        prescale                        (185),
        present-stage                   (493),
        present-value                   (85),
        …
        slave-proxy-enable              (172),
        stages                          (494),
        stage-names                     (495),
        start-time                      (142),
        ...
        tags                            (486),
        target-references               (496),
        threat-authority                (306),
        ...
-- -numerical order reference
        ...
        -- see represents              (491),
        -- see default-present-value    (492),
        -- see present-stage            (493),
        -- see stages                   (494),
        -- see stage-names              (495),
        -- see target-references        (496)
        ...
}
```

[Add to **Clause 21**, preserving the alphabetical order, p. 868]

BACnetStageLimitValue ::= SEQUENCE {

```
        limit                   REAL,
        values                  BIT STRING,     -- length is Nreferences bits, see Clause 12.X.6
        deadband                REAL
}
```

[Insert into **Table K-1** in **DS-V-A**, p. 1040]

Staging
Object_Name
Present_Value
Present_Stage
Status_Flags
Units

[Insert into **Table K-5** in **DS-M-A**, p. 1044]

Staging
Present_Value
Out_Of_Service

[Change Clause **K.1.27**, p. 1046]

K.1.27 BIBB - Data Sharing-Life Safety View-A (DS-LSV-A)

...

Devices claiming conformance to this BIBB shall be capable of reading and displaying the object properties listed in Table K-1, excluding properties of Averaging, Loop, Accumulator, Pulse Converter, Channel, Lighting Output, ~~and~~ Binary Lighting Output ~~objects~~, *and Staging objects*, and be capable of reading and displaying the object properties listed in Table K-7.

...

[Change Clause **K.1.28**, p. 1047]

K.1.28 BIBB - Data Sharing-Life Safety Advanced View-A (DS-LSAV-A)

The A device retrieves property values and presents them to the user. Device A shall be capable of using ReadProperty to retrieve any standard property of any standard object type listed in Table K-1, excluding Averaging, Loop, Accumulator, Pulse Converter, Channel, Lighting Output, ~~and~~ Binary Lighting Output ~~objects~~, *and Staging objects*, including the objects listed in Table K-7, except for those properties listed in Table K-2 and any property defined by the standard as not readable via ReadProperty. Device A may use alternate services where support for execution of the alternate service is supported by Device B.

...

[Change Clause **K.1.29**, p. 1047]

K.1.29 BIBB - Data Sharing-Life Safety Modify-A (DS-LSM-A)

...

Devices claiming conformance to this BIBB shall be capable of commanding and relinquishing standard commandable properties at priority 8 (other priorities may also be supported) of those objects listed in Table K-5 excluding Averaging, Loop, Accumulator, Pulse Converter, Channel, Lighting Output, ~~and~~ Binary Lighting Output ~~objects~~, *and Staging objects*, and writing the properties listed in Table K-5 and Table K-8, excluding Averaging, Loop, Accumulator, Pulse Converter, Channel, Lighting Output, ~~and~~ Binary Lighting Output ~~objects~~, *and Staging objects*.

...

[Change Clause **K.1.30**, p. 1048]

K.1.30 BIBB - Data Sharing-Life Safety Advanced Modify-A (DS-LSAM-A)

The A device is able to use WriteProperty to modify any standard property of object types listed in Tables K-5 and K-8, excluding Averaging, Loop, Accumulator, Pulse Converter, Channel, Lighting Output, ~~and~~ Binary Lighting Output ~~objects~~, *and Staging objects*, where the property is not required to be read-only, or to which access is otherwise restricted by the standard (e.g., Log_Buffer). Device A shall be capable of commanding and relinquishing standard commandable properties at any priority. Device A may use alternate services where support for execution of the alternate service is supported by Device B.

...

[Change Clause **K.1.31**, p. 1048]

K.1.31 BIBB - Data Sharing-Access Control View-A (DS-ACV-A)

...

Devices claiming conformance to this BIBB shall be capable of reading and displaying the object properties listed in Table K-1, excluding properties of Averaging, Loop, Accumulator, Pulse Converter, Channel, Lighting Output, ~~and~~ Binary Lighting Output ~~objects~~, *and Staging objects*, and be capable of reading and displaying the object properties listed in Table K-9.

...

[Change Clause **K.1.32**, p. 1049]

K.1.32 BIBB - Data Sharing-Access Control Advanced View-A (DS-ACAV-A)

The A device retrieves property values and presents them to the user. Device A shall be capable of using ReadProperty to retrieve any standard property of any standard object types listed in Table K-1, excluding Averaging, Loop, Accumulator, Pulse Converter, Channel, Lighting Output, ~~and~~ Binary Lighting Output ~~objects~~, *and Staging objects*, including the properties listed in Table K-9, except for those properties listed in Table K-2 and any property defined by the standard as not readable via ReadProperty. Device A may use alternate services where support for execution of the alternate service is supported by Device B.

...

[Change Clause **K.1.33**, p. 1050]

K.1.33 BIBB - Data Sharing-Access Control Modify-A (DS-ACM-A)

...

Devices claiming conformance to this BIBB shall be capable of commanding and relinquishing standard commandable properties at priority 8 (other priorities may also be supported) of those objects listed in Table K-5 excluding Averaging, Loop, Accumulator, Pulse Converter, Channel, Lighting Output, ~~and~~ Binary Lighting Output ~~objects~~, *and Staging objects*,

and writing the properties listed in Table K-5 and Table K-10, excluding Averaging, Loop, Accumulator, Pulse Converter, Channel, Lighting Output, ~~and~~ Binary Lighting Output ~~objects~~, *and Staging objects*.

[Change Clause **K.1.34**, p. 1050]

K.1.34 BIBB - Data Sharing–Access Control Advanced Modify-A (DS-ACAM-A)

The A device is able to use WriteProperty to modify any standard property of object types listed in Tables K-5 and K-10, excluding Averaging, Loop, Accumulator, Pulse Converter, Channel, Lighting Output, ~~and~~ Binary Lighting Output ~~objects~~, *and Staging objects*, where the property is not required to be read-only, or to which access is otherwise restricted by the standard (e.g., Log_Buffer). Device A shall be capable of commanding and relinquishing standard commandable properties at any priority. Device A may use alternate services where support for execution of the alternate service is supported by Device B.

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

(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.)

HISTORY OF REVISIONS

. . .	. . .	. . .
1	20	**Addendum *bd* to ANSI/ASHRAE Standard 135-2016** Approved by ASHRAE on June 15, 2018, and by the American National Standards Institute on June 15, 2018. 1. Add a Staging Object Type.

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.

ASHRAE · 1791 Tullie Circle NE · Atlanta, GA 30329 · www.ashrae.org

About ASHRAE

ASHRAE, founded in 1894, is a global society advancing human well-being through sustainable technology for the built environment. The Society and its members focus on building systems, energy efficiency, indoor air quality, refrigeration, and sustainability. Through research, Standards writing, publishing, certification and continuing education, ASHRAE shapes tomorrow's built environment today.

For more information or to become a member of ASHRAE, visit www.ashrae.org.

To stay current with this and other ASHRAE Standards and Guidelines, visit www.ashrae.org/standards.

Visit the ASHRAE Bookstore

ASHRAE offers its Standards and Guidelines in print, as immediately downloadable PDFs, on CD-ROM, 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.

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 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.

6/18