

BSR/ASHRAE Addendum *i*  
to ANSI/ASHRAE Standard 135-2008

# Public Review Draft

ASHRAE® Standard

## Proposed Addendum *i* to Standard 135-2008, *BACnet®—A Data Communication Protocol for Building Automation and Control Networks*

Third Public Review (**March 2009**)  
(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 addendum, go to the ASHRAE website at <http://www.ashrae.org/technology/page/331> 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 web site) remains in effect. The current edition of any standard may be purchased from the ASHRAE Bookstore @ <http://www.ashrae.org> or by calling 404-636-8400 or 1-800-527-4723 (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 web site @ <http://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 ASHRAE of any product, service, process, procedure, or design, and ASHRAE expressly disclaims such.

© **March 20, 2009**. 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, 1791 Tullie Circle, NE, Atlanta, GA 30329-2305. Phone: 404-636-8400, Ext1125. Fax: 404-321-5478. E-mail: [standards.section@ashrae.org](mailto:standards.section@ashrae.org).

AMERICAN SOCIETY OF HEATING,  
REFRIGERATING AND AIR-CONDITIONING  
ENGINEERS, INC.  
1791 Tullie Circle, NE · Atlanta GA 30329-2305



**[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.

SSPC 135 wishes to recognize the efforts of the following people in developing this addendum: David Fisher, Robert Hick, Pete Baselici, Steve Treado, Sharon Dinges, and the entire Lighting Applications working group. The committee is also grateful to Jörg Bröker, Christoph Zeller, John Hartman, Corey Balfour, Howard Coleman, John Fowler, and René Rauchenstein for their insightful comments.

135-2008i-1. Define new Lighting Output Object type, p. 1.

135-2008i-2. Add support for breaker-tripped status to Analog and Binary Output objects, p. 17.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2008 and Addenda is indicated through the use of *italics*, while deletions are indicated by ~~striketrough~~. Where entirely new subclauses are proposed to be added, plain type is used throughout. Only this new and deleted text is open to comment as 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.

**135-2008i-1. Define new Lighting Output Object type.**

**Rationale**  
A new object is needed to present the externally visible characteristics of a lighting output. Specifically, Lighting Outputs incorporate various features used in lighting applications, such as:  
- Blink Warning,  
- Continuous analog control of lighting level,  
- Ramping to a level at a fixed rate of change,  
- Fading to a level over a fixed period of time,  
- Incremental stepping values up and down,  
as well as a clear mapping to DALI-based lighting systems.

**Addendum 135-2008i-1**

[Insert new definitions in **Clause 3.2**, p. 2]

- ...  
**3.2.x fading:** the gradual increase or decrease of the actual output from one setting to another over a fixed period of time.
- ...  
**3.2.y ramping:** the gradual increase or decrease of the actual output from one setting to another at a fixed rate of change.
- ...  
**3.2.z stepping:** the increase or decrease of an output value in discrete steps.

[Change **Clause 12**, p. 129]

**12. MODELING CONTROL DEVICES AS A COLLECTION OF OBJECTS**

...  
Several object types defined in this clause have a property called "Reliability." This property is an enumerated datatype that may have different possible enumerations for different object types. The values defined below are a superset of all possible values of the Reliability property for all object types. The range of possible values returned for each specific object is defined in the appropriate object type definition.

...	
<b>TRIPPED</b>	<i>The actuator is not responding to commands, prevented by a tripped condition or by being mechanically held open.</i>
<b>UNRELIABLE_OTHER</b>	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.

[Add new **Clause 12.X**, p. 288]

**12.X Lighting Output Object Type**

The Lighting Output object type defines a standardized object whose properties represent the externally visible characteristics of a lighting output and includes dedicated functionality specific to lighting control that would otherwise require explicit programming. The lighting output can be analog or binary.

Lighting operations include ramping, stepping, and fading for dimming operations, delayed on or off, and blink warnings before off. The object includes current status indicators and statistics for tracking lamp life and usage.

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

**Table 12-X. Properties of the Lighting Output Object**

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	REAL	W
Progress_Value	REAL	R
Resolution	REAL	O
Binary_Present_Value	BACnetBinaryPV	O <sup>1</sup>
Output_Type	BACnetLightingOutputType	R
Lighting_Command	BACnetLightingCommand	O <sup>2</sup>
In_Process	BOOLEAN	R
Description	CharacterString	O
Fade_Time	Unsigned	O
Ramp_Rate	REAL	O
Blink_Enable	BOOLEAN	O <sup>3</sup>
Blink_Time	Unsigned	O <sup>3</sup>
Blink_Warn_Delay	Unsigned	O <sup>3</sup>
Step_Increment	REAL	O
Status_Flags	BACnetStatusFlags	R
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	R
Priority_Array	BACnetPriorityArray	R
Relinquish_Default	REAL	R
Power	REAL	O
Instantaneous_Power	REAL	O
Change_Of_State_Time	BACnetDateTime	O <sup>4</sup>
Change_Of_State_Count	Unsigned	O <sup>4</sup>
Time_Of_State_Count_Reset	BACnetDateTime	O <sup>4</sup>
Elapsed_Active_Time	Unsigned32	O <sup>5</sup>
Time_Of_Active_Time_Reset	BACnetDateTime	O <sup>5</sup>
Min_Pres_Value	REAL	O <sup>6</sup>
Max_Pres_Value	REAL	O <sup>6</sup>
Lighting_Command_Priority	Unsigned	R
Remote_Power_On_Value	BACnetOptionalREAL	O
Remote_System_Failure_Value	BACnetOptionalREAL	O
Binary_Active_Value	REAL	O <sup>1</sup>
Binary_Inactive_Value	REAL	O <sup>1</sup>
Polarity	BACnetPolarity	O <sup>1</sup>
Minimum_Off_Time	Unsigned32	O <sup>7</sup>

Minimum_On_Time	Unsigned32	O <sup>7</sup>
COV_Increment	REAL	O <sup>8</sup>
Profile_Name	CharacterString	O

- <sup>1</sup> If any of these properties are present, they shall all be present.
- <sup>2</sup> This property is required AND required to be writable if Output\_Type is not BINARY-LIGHTING.
- <sup>3</sup> These properties are required if the object supports warning blink.
- <sup>4</sup> If one of the optional properties Change\_Of\_State\_Time, Change\_Of\_State\_Count, or Time\_Of\_State\_Count\_Reset is present, then all of these properties shall be present.
- <sup>5</sup> If one of the optional properties Elapsed\_Active\_Time or Time\_Of\_Active\_Time\_Reset is present, then both of these properties shall be present.
- <sup>6</sup> If either of these properties is present, they shall both be present, and they are required to be writable.
- <sup>7</sup> If either of these properties is present, they shall both be present.
- <sup>8</sup> This property is required if the object supports COV reporting.

**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 LIGHTING\_OUTPUT.

**12.X.4 Present\_Value (Commandable)**

This property, of type REAL, indicates the linearized percentage (0..100%) of the device’s target light output range (LUMENS), with 0 being off, 1 being dimmest, 100 being brightest. If the object supports fading (indicated by the presence of Fade\_Time), or ramping (indicated by the presence of Ramp\_Rate), or delaying (indicated by the presence of Blink\_Warn\_Delay), then it is possible that Present\_Value may not indicate the actual state of the lighting output due to a fade, ramp or delay in progress. If the Min\_Pres\_Value property is present, then the Present\_Value (when greater than zero) shall be forced to be no less than Min\_Pres\_Value. If the Max\_Pres\_Value property is present, then the Present\_Value (when greater than zero) shall be forced to be no greater than Max\_Pres\_Value. Present\_Value shall always be a non-negative number in the range 0 to 100%. Writes to Present\_Value less than 1 and greater than 0 shall be clamped to 1, or Min\_Pres\_Value if present. Writes to Present\_Value of values below 0 or above 100 shall cause a Result(-) to be returned with an error class of PROPERTY and an error code of VALUE\_OUT\_OF\_RANGE.

Writes to the Lighting\_Command property may initiate ramping, fading or stepping operations that asynchronously affect the lighting output by establishing a new target for Present\_Value and carrying out the requested operation. New light levels due to a command to the Present\_Value property are reached by a single step to the new value. Fading and ramping are supported for commands to the property Lighting\_Command but not for commands to the property Present\_Value.

**12.X.5 Progress\_Value**

This property, of type REAL, indicates the linearized percentage (0..100%) of the device’s actual light output range (LUMENS); 0 being off, 1 being dimmest, 100 being brightest. If the object supports fading (indicated by the presence of Fade\_Time) or ramping (indicated by the presence of Ramp\_Rate), then Progress\_Value shall indicate the actual state of the lighting output while a fade or ramp is in progress. The manner by which the Progress\_Value is determined shall be a local matter.

**12.X.6 Resolution**

This optional property, of type REAL, indicates the smallest recognizable change in Present\_Value in engineering units (read-only). Some lighting outputs (such as those representing DALI ballasts) may have a non-linear resolution.

**12.X.7 Binary\_Present\_Value**

This optional property, of type BACnetBinaryPV, shall be in either of two states: INACTIVE or ACTIVE. At power-on the state shall be INACTIVE. Thereafter, if the value of the Present\_Value property becomes greater than or equal to Binary\_Active\_Value, then Binary\_Present\_Value shall become ACTIVE. Once ACTIVE, the Binary\_Present\_Value shall remain ACTIVE until Present\_Value becomes less than or equal to Binary\_Inactive\_Value, at which point Binary\_Present\_Value shall become INACTIVE. Binary\_Present\_Value shall be read-only.

**12.X.8 Output\_Type**

This property, of type BACnetLightingOutputType, shall indicate whether the underlying Lighting Output is a binary or an analog type of lighting output.

**12.X.9 Lighting\_Command**

This property, of type BACnetLightingCommand, is used to request special lighting commands with specific behaviors. Lighting\_Command is written with compound values that specify particular lighting operations. Devices are not required to support all BACnetLightingOperations. At a minimum, devices containing Lighting Output objects shall support STOP, GOTO and RELINQUISH. If Output\_Type is any value other than BINARY\_LIGHTING, then all BACnetLightingOperations shall be supported. For BINARY\_LIGHTING types of Lighting Output objects, the interpretation of each lighting operation shall be a local matter. If Lighting\_Command is written with a value that specifies an unsupported operation, the write shall fail and a Result(-) returned with an error class of PROPERTY and an error code of OPTIONAL\_FUNCTIONALITY\_NOT\_SUPPORTED. If another lighting operation is in progress when a successful write occurs, the previous operation shall be stopped. The fields of the BACnetLightingCommand are summarized in Table 12-Y.

**Table 12-Y.** LightingCommand Property Fields

Field	Description
operation	an enumeration of type BACnetLightingOperation that defines the operation desired
level	the target level for certain fade or ramp operations
ramp-rate	the rate of change in percent-per-second for certain ramp operations
step-increment	the amount to be added to or subtracted from Present_Value when stepping
fade-time	the time in milliseconds over which certain fade operations take place with a minimum resolution of 100 ms
duration	an overall time duration in milliseconds, with a minimum resolution of 100 ms, after which the target level Priority_Array slot is relinquished automatically

When lighting operations occur, they affect the Present\_Value at the priority level specified in the Lighting\_Command\_Priority property. The writing of the Lighting\_Command property shall stop any lighting operation in progress at the time that the write occurs for any supported lighting command. Lighting operations that include a target level shall save the new level ("the target level") in the Priority\_Array slot specified by the Lighting\_Command\_Priority property. Fades and ramps to new target values shall be calculated from actual present lighting output (represented by the Progress\_Value) to preserve a smooth transition of lighting level. The calculated level from moment to moment shall be called the "calculated level." The lighting operations that cause incremental effects on Present\_Value (ramping, fading and stepping) shall be clamped to a minimum of Min\_Pres\_Value and a maximum of Max\_Pres\_Value respectively when both are present. Clamping shall occur prior to initiation of ramping, fading or stepping to assure that only a clamped target value is used.

Some lighting operations require additional parameters. These are provided by optional fields of the BACnetLightingCommand value. Any lighting command operation can specify a time duration in seconds after which the target level Priority\_Array slot is relinquished automatically. If duration is not specified, then no automatic relinquish shall be assumed and any previously specified duration shall remain in effect. If a duration is specified and is greater

than zero, it shall reset any previously specified duration that is currently in effect. If a duration of zero is specified, it shall stop any duration currently in effect, cancelling the automatic relinquish.

Lighting operations that specify a rate of change, or a time duration, shall be limited to an expected timing resolution of 100ms or better. Some lighting devices may incorporate remote subnetworks or other technology that may introduce latency or non-linearity in the behavior of the physical light being controlled. Consequently, the absolute timing resolution of lighting operations should not be assumed. Some lighting devices may not be capable of achieving the performance implied by a given operation, in which case the device shall use its best effort to carry out the intended operation.

The lighting operations shall be represented by one of the following enumerated values:

{STOP, FADE\_TO, FADE\_TO\_OVER, RAMP\_TO, RAMP\_TO\_AT\_RATE, STEP\_UP, STEP\_DOWN, STEP\_UP\_BY, STEP\_DOWN\_BY, GOTO, RELINQUISH}

The Lighting\_Command property shall indicate the last commanded value, or STOP if it has not been commanded.

The BACnetLightingOperations are summarized in the following table. Optional fields of the BACnetLightingOperation value are shown in **bold**:

**Table 12-Z.** BACnetLightingOperations

Operation	Description
STOP	stops any fade or ramp in progress.
FADE_TO	<b>level</b> is the target level. The calculated level is changed proportionally from a starting value until it reaches <b>level</b> over a period defined by the Fade_Time property.
FADE_TO_OVER	<b>level</b> is the target level. The calculated level is changed proportionally from its starting value until it reaches <b>level</b> over a period defined by <b>fade-time</b> .
RAMP_TO	<b>level</b> is the target level. The calculated level is changed by adding or subtracting an amount, proportional over time according to the Ramp_Rate property, to/from the starting level until it reaches <b>level</b> .
RAMP_TO_AT_RATE	<b>level</b> is the target level. The calculated level is changed by adding or subtracting an amount, proportional over time according to <b>ramp-rate</b> , to/from the starting level until it reaches <b>level</b> .
STEP_UP	The calculated level is increased from the starting level by adding an amount according to the Step_Increment property.
STEP_DOWN	The calculated level is decreased from the starting level by subtracting an amount according to the Step_Increment property.
STEP_UP_BY	The calculated level is increased from the starting level by adding an amount according to the <b>step-increment</b> .
STEP_DOWN_BY	The calculated level is decreased from the starting level by subtracting an amount according to the <b>step-increment</b> .
GOTO	<b>level</b> is the target level.
RELINQUISH	Causes the target level Priority_Array slot to be relinquished at the priority specified by Lighting_Command_Priority.

#### 12.X.10 In\_Process

This property, of type BOOLEAN, shall be TRUE whenever a lighting command is being executed as a result of a write to the Lighting\_Command property. When the calculated level reaches the target level, In\_Process shall become FALSE.

### 12.X.11 Description

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

### 12.X.12 Fade\_Time

This optional property, of type `Unsigned`, indicates the amount of time in milliseconds over which changes to the actual value of the lighting output shall occur when the `Lighting_Command` property is written with a fade request that does not include a fade-time value. The minimum resolution of `Fade_Time` shall be 100 ms.

### 12.X.13 Ramp\_Rate

This optional property, of type `REAL`, indicates the rate in percent per second at which changes to the actual value of the lighting output shall occur when the `Lighting_Command` property is written with a ramp request that does not include a ramp-rate value.

### 12.X.14 Blink\_Enable

This optional property, of type `BOOLEAN`, indicates whether blink warning functionality is enabled (`TRUE`) or disabled (`FALSE`). If `Blink_Enable` is `TRUE`, then the functionality of `Minimum_Off_Time` and `Minimum_On_Time` shall be disabled if those properties are present.

### 12.X.15 Blink\_Time

This optional property, of type `Unsigned`, indicates the time in milliseconds that the output shall blink off when `Present_Value` becomes a 0% value (a blink warning) AND `Blink_Enable` is `TRUE`. A `Blink_Time` of zero shall disable this blink-off effect. If the `Blink_Time` property is present, then the `Blink_Warn_Delay` property shall also be present. The minimum resolution of `Blink_Time` shall be 100 ms.

#### 12.X.15.1 Blink Warning Behavior

If `Blink_Time` is non-zero, then blink warning shall be enabled. When `Blink_Time` is non-zero and `Present_Value` becomes 0% at a priority of `X`, then if all of the `Priority_Array` slot numbers are greater than `X` and `Relinquish_Default` are 0 or `NULL`, then the output goes to zero (blinking off) for the time specified by `Blink_Time`. After the time specified by `Blink_Time` has elapsed, the output is restored to its previous state and may remain on or blink repeatedly on and off for up to `Blink_Warn_Delay` milliseconds. When `Blink_Time` is zero and `Present_Value` becomes 0%, the lights shall remain on and not blink for up to `Blink_Warn_Delay` milliseconds. Whether there is a single blink or multiple blinks shall be a local matter. If `Blink_Warn_Delay` is non-zero, then the `Priority_Array` slot that corresponds to the priority at which `Present_Value` was written with 0% shall be automatically relinquished at the end of the `Blink_Warn_Delay`. If `Present_Value` is written at a priority lower than 6, then the blink shall not occur and the lower numbered priority slot shall not be relinquished. If there is a blink or blink warn delay in effect when `Present_Value` is written, then when `Present_Value` is written with a zero, the blink time shall be relinquished immediately at priority 6 and the priority at which the zero was written. If `Present_Value` is written with a non-zero value, then the blink time shall be relinquished immediately at priority 6. These procedures may be more clearly stated programmatically:

```
if Present_Value is written with 0 at a priority of X then
  if a blink time or blink warn delay is in effect then
    object internally writes NULL at priority 6
    object internally writes NULL at priority X to auto relinquish
  else
    if priority X > 6 then
      if Relinquish_Default and all Priority_Array[i] for i=7 to 16 are ==0 or NULL then
        Y=Present_Value
        object internally writes 0 at priority 6
        wait Blink_Time
        object internally writes Y at priority 6
        while waiting for Blink_Warn_Delay
          (local choice) do nothing or alternately write 0 or Y at priority 6 to blink
            on/off/on/off
        endwhile
        object internally writes NULL at priority 6
      endif
      object internally writes NULL at priority X to auto relinquish
    else
      // priority X was <=6
      just write 0 at priority X
    endif
  endif
```

```
else
    if a blink time or blink warn delay is in effect then object internally writes NULL at priority 6
    just write the value at priority X
endif
```

#### 12.X.16 Blink\_Warn\_Delay

This optional property, of type Unsigned, indicates the time in milliseconds after a blink warning until the Present\_Value is automatically relinquished. A Blink\_Warn\_Delay of zero disables this automatic relinquish. If the Blink\_Warn\_Delay property is present, then the Blink\_Time property shall also be present and the Minimum\_Off\_Time and Minimum\_On\_Time properties shall not be present. The minimum resolution of Blink\_Warn\_Delay shall be 100 ms.

#### 12.X.17 Step\_Increment

This optional property, of type REAL, indicates the amount to be added to or subtracted from the starting level when the Lighting\_Command property is written with a step request that does not include a step increment value.

#### 12.X.18 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 Always Logical FALSE (0).

FAULT Logical TRUE (1) if the Reliability property is present and does not have a value of NO\_FAULT\_DETECTED, otherwise logical FALSE (0).

OVERRIDDEN 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.

OUT\_OF\_SERVICE Logical TRUE (1) if the Out\_Of\_Service property has a value of TRUE, otherwise logical FALSE (0).

#### 12.X.19 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the Present\_Value or the operation of the physical output in question is "reliable" as far as the BACnet Device or operator can determine and, if not, why. The Reliability property for this object type may have any of the following values:

{NO\_FAULT\_DETECTED, OPEN\_LOOP, SHORTED\_LOOP, NO\_OUTPUT, COMMUNICATION\_FAILURE, TRIPPED, UNRELIABLE\_OTHER}

#### 12.X.20 Out\_Of\_Service

The Out\_Of\_Service property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the physical point that the object represents is not in service. This means that changes to the Present\_Value property are decoupled from the physical output when the value of Out\_Of\_Service is TRUE. In addition, the Reliability property and the corresponding state of the FAULT flag of the Status\_Flags property shall be decoupled from the physical output when Out\_Of\_Service is TRUE. While the Out\_Of\_Service property is TRUE, the Present\_Value and Reliability properties may still be changed to any value as a means of simulating specific fixed conditions or for testing purposes. Other functions that depend on the state of the Present\_Value or Reliability properties shall respond to changes made to these properties while Out\_Of\_Service is TRUE, as if those changes had occurred to the physical output. The Present\_Value property shall still be controlled by the BACnet command prioritization mechanism if Out\_Of\_Service is TRUE. See Clause 19.

#### **12.X.21 Priority\_Array**

This property is a read-only array of prioritized values. See Clause 19 for a description of the prioritization mechanism.

#### **12.X.22 Relinquish\_Default**

This property, of type REAL, is the default value to be used for the Present\_Value property when all command priority values in the Priority\_Array property have a NULL value. See Clause 19.

#### **12.X.23 Power**

This optional property, of type REAL, is the nominal power consumption of the load(s) controlled by this object when 100% ON. The units shall be kilowatts.

#### **12.X.24 Instantaneous\_Power**

This optional property, of type REAL, is the nominal power consumption of the load(s) controlled by this object at this moment. The units shall be kilowatts.

#### **12.X.25 Change\_Of\_State\_Time**

This optional property, of type BACnetDateTime, represents the date and time at which the most recent change of state occurred. A "change of state" shall be defined as any event that alters the Present\_Value property from either the Binary\_Active\_Value property to the Binary\_Inactive\_Value property, or the Binary\_Inactive\_Value property to the Binary\_Active\_Value property (if present), or zero to non-zero, or non-zero to zero. If any of the optional properties Change\_Of\_State\_Time, Change\_Of\_State\_Count, or Time\_Of\_State\_Count\_Reset is present, then all of these properties shall be present.

#### **12.X.26 Change\_Of\_State\_Count**

This optional property, of type Unsigned, represents the number of times that the Present\_Value property has changed state since the Change\_Of\_State\_Count property was most recently set to a zero value. The Change\_Of\_State\_Count property shall have a range of 0-65535 or greater. A "change of state" shall be defined as any event that alters the Present\_Value property from either the Binary\_Active\_Value property to the Binary\_Inactive\_Value property, or the Binary\_Inactive\_Value property to the Binary\_Active\_Value property (if present), or zero to non-zero, or non-zero to zero. If any of the optional properties Change\_Of\_State\_Time, Change\_Of\_State\_Count, or Time\_Of\_State\_Count\_Reset is present, then all of these properties shall be present.

#### **12.X.27 Time\_Of\_State\_Count\_Reset**

This optional property, of type BACnetDateTime, represents the date and time at which the Change\_Of\_State\_Count property was most recently set to a zero value. If any of the optional properties Change\_Of\_State\_Time, Change\_Of\_State\_Count, or Time\_Of\_State\_Count\_Reset is present, then all of these properties shall be present.

#### **12.X.28 Elapsed\_Active\_Time**

This optional property, of type Unsigned32, represents the accumulated number of seconds that the Present\_Value has been ACTIVE. This value is reset when a value of zero is written to this property. ACTIVE shall be defined as the Binary\_Active\_Value property if it is present, or as a non-zero value if the Binary\_Active\_Value property is not present. If one of the optional properties Elapsed\_Active\_Time or Time\_Of\_Active\_Time\_Reset is present, then both of these properties shall be present.

#### **12.X.29 Time\_Of\_Active\_Time\_Reset**

This optional property, of type BACnetDateTime, represents the date and time at which the Elapsed\_Active\_Time property was most recently set to a zero value. If one of the optional properties Elapsed\_Active\_Time or Time\_Of\_Active\_Time\_Reset is present, then both of these properties shall be present.

#### **12.X.30 Min\_Pres\_Value**

This optional property, of type REAL, shall specify the minimum non-zero value to which Present\_Value shall be restricted. Changing Present\_Value to a value greater than zero and below Min\_Pres\_Value shall force Present\_Value to be Min\_Pres\_Value. Changing Min\_Pres\_Value to a value greater than Present\_Value shall force Present\_Value to become Min\_Pres\_Value. If present, the Min\_Pres\_Value property shall be required to be writable and Max\_Pres\_Value shall also be present. Min\_Pres\_Value shall always be a positive number in the range 1 to 100%.

**12.X.31 Max\_Pres\_Value**

This optional property, of type REAL, shall specify the maximum non-zero value to which Present\_Value shall be restricted. Changing Present\_Value to a value greater than zero and above Max\_Pres\_Value shall force Present\_Value to be Max\_Pres\_Value. Changing Max\_Pres\_Value to a value less than Present\_Value shall force Present\_Value to become Max\_Pres\_Value. If present, the Max\_Pres\_Value property shall be required to be writable and Min\_Pres\_Value shall also be present. Max\_Pres\_Value shall always be a positive number in the range 1 to 100%.

**12.X.32 Lighting\_Command\_Priority**

This property, of type Unsigned, shall specify a write priority of 1 to 16 that indicates the element of the Priority\_Array controlled by the Lighting\_Command property.

**12.X.33 Remote\_Power\_On\_Value**

This optional property, of type BACnetOptionalREAL, shall specify the level that the physical output shall go to after power is applied in those cases when the physical output is controlled by a device that is physically separate from the device containing the Lighting Output object. A value of NULL shall specify that the physical output is set to the value it had before the power was turned off or failed.

**12.X.34 Remote\_System\_Failure\_Value**

This optional property, of type BACnetOptionalREAL, shall specify the value that a physical output shall go to if the interface between the Lighting Output object and the output should fail in those cases when the physical output is controlled by a device that is physically separate from the device containing the Lighting Output object. A value of NULL shall specify that the physical output remain in the state that it is in.

**12.X.35 Binary\_Active\_Value**

This optional property, of type REAL, is the value of the Present\_Value property at or above which a binary actuator (i.e., relay) becomes ACTIVE. The relationship between the Present\_Value and the physical state of the output is determined by the Polarity property. The possible states are summarized in Table 12-A. If any of the optional properties Binary\_Active\_Value, Binary\_Inactive\_Value, or Polarity is present, then all of these properties shall be present.

**12.X.36 Binary\_Inactive\_Value**

This optional property, of type REAL, is the value of the Present\_Value property at or below which a binary actuator (i.e., relay) becomes INACTIVE. The relationship between the Present\_Value and the physical state of the output is determined by the Polarity property. The possible states are summarized in Table 12-A. If any of the optional properties Binary\_Active\_Value, Binary\_Inactive\_Value, or Polarity is present, then all of these properties shall be present.

**12.X.37 Polarity**

This optional property, of type BACnetPolarity, indicates the relationship between the physical state of the output and the logical state represented by the Present\_Value property as interpreted by the Binary\_Active\_Value and Binary\_Inactive\_Value properties. If the Polarity property is NORMAL, then the ACTIVE state of the Present\_Value property is also the ACTIVE or ON state of the physical output as long as Out\_Of\_Service is FALSE. If the Polarity property is REVERSE, then the ACTIVE state of the Present\_Value property is the INACTIVE or OFF state of the physical output as long as Out\_Of\_Service is FALSE. See Table 12-A. If Out\_Of\_Service is TRUE, then the Polarity property shall have no effect on the physical output state. If any of the optional properties Binary\_Active\_Value, Binary\_Inactive\_Value, or Polarity is present, then all of these properties shall be present.

**Table 12-A. BACnet Polarity Relationships**

Present_Value	Polarity	Physical State of Output
<b>&gt;= Binary_Active_Value</b>	NORMAL	ON or ACTIVE
<b>&lt;= Binary_Inactive_Value</b>	NORMAL	OFF or INACTIVE
<b>&gt;= Binary_Active_Value</b>	REVERSE	OFF or INACTIVE
<b>&lt;= Binary_Inactive_Value</b>	REVERSE	ON or ACTIVE

#### **12.X.38 Minimum\_Off\_Time**

This optional property, of type Unsigned32, represents the minimum number of seconds that the Present\_Value shall remain in the INACTIVE state after a write to the Present\_Value property causes that property to assume the INACTIVE state. INACTIVE shall be defined as the Binary\_Inactive\_Value property if it is present, or a value of zero if the Binary\_Inactive\_Value property is not present. The mechanism by which this is accomplished is a local matter. If this property is present then Minimum\_On\_Time shall also be present.

#### **12.X.39 Minimum\_On\_Time**

This optional property, of type Unsigned32, represents the minimum number of seconds that the Present\_Value shall remain in the ACTIVE state after a write to the Present\_Value property causes that property to assume the ACTIVE state. ACTIVE shall be defined as the Binary\_Active\_Value property if it is present, or as a non-zero value if the Binary\_Active\_Value property is not present. The mechanism by which this is accomplished is a local matter. If this property is present then Minimum\_Off\_Time shall also be present

#### **12.X.40 COV\_Increment**

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

#### **12.X.41 Profile\_Name**

This optional property, of type CharacterString, is the name of an object profile to which this object conforms. To ensure uniqueness, a profile name must 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 document named by the remainder of the profile name. 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. The definition of the profiles themselves is outside the scope of this standard.

[Change **Table 13-1**, p. 291]

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

Object Type	Criteria	Properties Reported
Analog Input, Analog Output, Analog Value, <i>Lighting Output</i>	If Present_Value changes by COV_Increment or Status_Flags changes at all	Present_Value, Status_Flags
Binary Input, Binary Output, Binary Value, Life Safety Point, Life Safety Zone, Multi-state Input, Multi-state Output, Multi-state Value	If Present_Value changes at all or Status_Flags changes at all	Present_Value, Status_Flags
Loop	If Present_Value changes by COV_Increment or Status_Flags changes at all	Present_Value, Status_Flags, Setpoint, Controlled_Variable_Value
Pulse Converter	If Present_Value changes by COV_Increment or Status_Flags changes at all or If COV_Period expires	Present_Value, Status_Flags, Update_Time

[Change **Clause 19.2.3**, p.403]

**19.2.3 Minimum\_On\_Time and Minimum\_Off\_Time**

If the commandable property is the Present\_Value property of a Binary Output object or a Binary Value object and that object possesses the optional Minimum\_On\_Time and Minimum\_Off\_Time properties, then minimum on and minimum off times shall behave according to the algorithm described in this subclause.

Command priority 6 is reserved for use by this algorithm *and by the Blink Warning algorithm in the Lighting Output object*, and may not be used for other purposes in any object.

[Add to **Clause 21**, new **BACnetLightingCommand** production, p. 459]

-- Note that the combination of level, ramp-rate, step-increment and fade-time fields is  
-- dependent on the specific lighting operation (see Table 12-Y).

```
BACnetLightingCommand ::= SEQUENCE {  
    operation          [0] BACnetLightingOperation,  
    level              [1] REAL OPTIONAL,  
    ramp-rate          [2] REAL OPTIONAL,  
    step-increment     [3] REAL OPTIONAL,  
    fade-time          [4] Unsigned OPTIONAL,  
    duration           [5] Unsigned OPTIONAL  
}
```

[Add to **Clause 21**, new **BACnetLightingOperation** production, p. 459]

```
BACnetLightingOperation ::= ENUMERATED {  
    stop                (0),  
    fade-to             (1),  
    fade-to-over        (2),  
    ramp-to             (3),  
    ramp-to-at-rate     (4),  
    step-up             (5),  
    step-down           (6),  
    step-up-by          (7),  
    step-down-by        (8),  
    goto                (9),  
    relinquish          (10)  
}
```

-- Enumerated values 0-255 are reserved for definition by ASHRAE. Enumerated values 256-65535 may be used by  
-- others subject to the procedures and constraints described in Clause 23.

[Add to **Clause 21**, new **BACnetLightingOutputType** production, p. 459]

```
BACnetLightingOutputType ::= ENUMERATED {  
    binary-lighting     (0),  
    analog-lighting     (1),  
    analog-color-lighting (2) -- reserved for anticipated color lighting  
}
```

[Add to **Clause 21**, new **BACnetOptionalReal** production, p. 464]

```
BACnetOptionalREAL ::= CHOICE {  
    real-value          REAL,  
    default-value      NULL  
}
```

[Change **Clause 21**, existing **BACnetObjectType** enumeration, p. 463]

[Note: enumerations 25 through 30 are used in Addenda *b-1*, *d-1*, *e-1*, and *f-1* to ANSI/ASHRAE 135-2004.]

```
BACnetObjectType ::= ENUMERATED {  
  ...  
    life-safety-zone          (22),  
    lighting-output          (x),  
    loop                      (12),  
    ...  
    -- see lighting-output    (x),  
    ...  
  }
```

[Change **Clause 21**, existing **BACnetObjectTypesSupported** production, p. 463]

```
BACnetObjectTypesSupported ::= BIT STRING {  
  ...  
  -- life-safety-zone        (22),  
  -- lighting-output         (x),  
  loop                      (12),  
  ...,  
  lighting-output           (x)  
  }
```

[Change **Clause 21**, existing **BACnetPropertyIdentifier** production , p. 465]

```
BACnetPropertyIdentifier ::= ENUMERATED {  
  ...  
  bias                      (14),  
  binary-active-value       (x),  
  binary-inactive-value     (x+1),  
  blink-enable              (x+2),  
  blink-time                (x+3),  
  blink-warn-delay          (x+4),  
  buffer-size               (126),  
  ...  
  exception-schedule        (38),  
  fade-time                 (x+5),  
  fault-values              (39),  
  ...  
  life-safety-alarm-values  (166),  
  lighting-command          (x+6),  
  lighting-command-priority (x+7),  
  limit-enable              (52),  
  ...  
  operation-expected        (161),  
  output-type               (x+8),  
  ...  
  polarity                  (84),  
  power                     (x+9),  
  prescale                  (185),  
  ...  
  program-state             (92),  
  progress-value            (x+10),  
  proportional-constant     (93),  
  ...
```

```

protocol-version          (98),
ramp-rate                 (x+11),
read-only                 (99),
remote-power-on-value    (x+12),
remote-system-failure-value (x+13),
...
status-flags             (111),
step-increment           (x+14),
stop-time                (143),
...
-- see value-change-time (192),
...
-- see binary-active-value (x),
-- see binary-inactive-value (x+1),
-- see blink-enable (x+2),
-- see blink-time (x+3),
-- see blink-ward-delay (x+4),
-- see fade-time (x+5),
-- see lighting-command (x+6),
-- see lighting-command-priority (x+7),
-- see output-type (x+8),
-- see power (x+9),
-- see progress-value (x+10),
-- see ramp-rate (x+11),
-- see remote-power-on-value (x+12),
-- see remote-system-failure-value (x+13),
-- see step-increment (x+14),
}
    
```

[Change **Clause 21**, existing **BACnetReliability** enumeration, p. 472]

**BACnetReliability** ::= ENUMERATED {

```

...
Tripped (x),
...
}
    
```

-- 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.

[Change **Clause 23.1**, p. 480]

**23.1 Extending Enumeration Values**

There may be instances when it is necessary for a vendor to extend BACnet by including additional possible values to an enumeration. This is accomplished by using enumeration values that are greater than the range reserved for BACnet for a given enumeration type. Table 23-1 defines those enumerations that may be extended and the range of enumerated values reserved for BACnet use. All other enumerations, which do not appear in Table 23-1, may not be extended.

**Table 23-1. Extensible Enumerations**

Enumeration Name	Reserved Range	Maximum Value
...	...	...
BACnetLifeSafetyOperation	0-63	65535
BACnetLightingOperation	0-255	65535
...	...	...

[Add to **Annex C**, p. 460]

```

LIGHTING-OUTPUT ::= SEQUENCE {
    object-identifier [75] BACnetObjectIdentifier,
    object-name [77] CharacterString,
    object-type [79] BACnetObjectType,
    present-value [85] REAL,
    progress-value [x+10] REAL,
    resolution [106] REAL OPTIONAL,
    lighting-command [x+6] BACnetLightingCommand,
    in-process [47] BOOLEAN OPTIONAL,
    description [28] CharacterString OPTIONAL,
    fade-time [x+5] Unsigned OPTIONAL,
    ramp-rate [x+11] REAL OPTIONAL,
    blink-enable [x+2] BOOLEAN,
    blink-time [x+3] Unsigned OPTIONAL,
    blink-warn-delay [x+4] Unsigned OPTIONAL,
    step-increment [x+14] REAL OPTIONAL,
    status-flags [111] BACnetStatusFlags,
    reliability [103] BACnetReliability OPTIONAL,
    out-of-service [81] BOOLEAN,
    priority-array [87] BACnetPriorityArray,
    relinquish-default [104] REAL,
    power [x+9] REAL OPTIONAL,
    change-of-state-time [16] BACnetDateTime OPTIONAL,
    change-of-state-count [15] Unsigned OPTIONAL,
    time-of-state-count-reset [115] BACnetDateTime OPTIONAL,
    elapsed-active-time [33] Unsigned32 OPTIONAL,
    time-of-active-time-reset [114] BACnetDateTime OPTIONAL,
    min-pres-value [69] REAL OPTIONAL,
    max-pres-value [65] REAL OPTIONAL,
    lighting-command-priority [x+7] Unsigned,
    remote-power-on-value [x+12] BACnetOptionalREAL OPTIONAL,
    remote-system-failure-value [x+13] BACnetOptionalREAL OPTIONAL,
    binary-active-value [x] REAL OPTIONAL,
    binary-inactive-value [x+1] REAL OPTIONAL,
    polarity [84] BACnetPolarity OPTIONAL,
    minimum-off-time [66] Unsigned32 OPTIONAL,
    minimum-on-time [67] Unsigned32 OPTIONAL,
    cov-increment [22] REAL OPTIONAL,
    profile-name [167] CharacterString OPTIONAL
}

```

[Add new **Annex D.X**, p.484]

#### **D.X Example of a Lighting Output Object**

Property:	Object_Identifier =	(Lighting Output, Instance 1)
Property:	Object_Name =	"ConferenceRoom"
Property:	Object_Type =	LIGHTING_OUTPUT
Property:	Present_Value =	72.00%
Property:	Progress_Value =	68.25%
Property:	Lighting_Command =	{FADE_TO_OVER,30,20,60} -- Fade to 30% over 20 seconds and relinquish after 60 seconds
Property:	In_Process =	TRUE
Property:	Description =	"Some Description"
Property:	Fade_Time =	100000

Property:	Ramp_Rate =	0.0
Property:	Blink_Enable =	TRUE
Property:	Blink_Time =	700
Property:	Blink_Warn_Delay =	120000
Property:	Step_Increment =	5.0
Property:	Status_Flags =	{FALSE, FALSE, FALSE, FALSE}
Property:	Reliability =	NO_FAULT_DETECTED
Property:	Out_Of_Service =	FALSE
Property:	Priority_Array =	{NULL ,NULL, NULL, NULL, 72.0... NULL}
Property:	Relinquish_Default =	0.0
Property:	Power =	1.5
Property:	Instantaneous_Power =	1.02375
Property:	Change_Of_State_Time =	(23-MAR-1995,19:01:34.0)
Property:	Change_Of_State_Count =	47
Property:	Time_Of_State_Count_Reset =	(1-JAN-1995,00:00:00.0)
Property:	Elapsed_Active_Time =	650
Property:	Time_Of_Active_Time_Reset =	(1-JAN-1995,00:00:00.0)
Property:	Min_Pres_Value =	10.0
Property:	Max_Pres_Value =	100.0
Property:	Lighting_Command_Priority =	10
Property:	Remote_Power_On_Value =	0.0
Property:	Remote_System_Failure_Value =	NULL
Property:	Binary_Active_Value =	50.0
Property:	Binary_Inactive_Value =	45.0
Property:	Polarity =	NORMAL
Property:	Minimum_Off_Time =	600
Property:	Minimum_On_Time =	0

**135-2008i-2. Add support for breaker-tripped status to Analog and Binary Output objects.**

**Rationale**

"Breaker-tripped" status does not map to any of the existing Analog and Binary Output Reliability property enumerations.

**Addendum 135-2008i-2**

[Change **Clause 12.3.9**, p. 146]

**12.3.9 Reliability**

The Reliability property, of type BACnetReliability, provides an indication of whether the Present\_Value or the operation of the physical output in question is "reliable" as far as the BACnet Device or operator can determine and, if not, why. The Reliability property for this object type may have any of the following values:

{NO\_FAULT\_DETECTED, OPEN\_LOOP, SHORTED\_LOOP, NO\_OUTPUT, *TRIPPED*, UNRELIABLE\_OTHER}

[Change **Clause 12.7.9**, p. 163]

**12.7.9 Reliability**

The Reliability property, of type BACnetReliability, provides an indication of whether the Present\_Value or the operation of the physical output in question is "reliable" as far as the BACnet Device or operator can determine and, if not, why. The Reliability property for this object type may have any of the following values:

{NO\_FAULT\_DETECTED, NO\_OUTPUT, OPEN\_LOOP, SHORTED\_LOOP, *TRIPPED*, UNRELIABLE\_OTHER}.