

BSR/ASHRAE Addendum *i*
to ANSI/ASHRAE Standard 135-2004

Public Review Draft

ASHRAE® Standard

Proposed Addendum *i* to Standard 135-2004, *BACnet®—A Data Communication Protocol for Building Automation and Control Networks*

Second Public Review (**March 2008**)
(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, use the comment form and instructions provided with this draft. 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 25, 2008**. 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. Phone: 404-636-8400, Ext. 1125. Fax: 404-321-5478. E-mail: 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.

135-2004*i*-1. Define new Lighting Output Object type, p. 1.

135-2004*i*-2. Add support for breaker-tripped status to Analog and Binary Output objects, p. 15.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2004 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 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.

Note to Reviewers: This draft was first posted on March 21, 2008, with a typographical error in Section 12.X.36. The error has been corrected in this draft and this corrected draft was re-posted for public review on March 25, 2008. The public review period has been extended to May 9, 2008 for this addendum.

135-2004i-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 135-2004i-1

[Insert new **3.2.21**, p. 3, renumbering subsequent clauses]

3.2.21 fading: the gradual increase or decrease of the actual output from one setting to another over a fixed period of time.

[Insert new **3.2.45**, p. 4, (following insertion of 3.2.21) renumbering subsequent clauses]

3.2.45 ramping: the gradual increase or decrease of the actual output from one setting to another at a fixed rate of change.

[Insert new **3.2.57**, p. 5, (following insertion of 3.2.45) renumbering subsequent clauses]

3.2.57 stepping: the increase or decrease of an output value in discrete steps.

[Change **12**, pp. 127-129]

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.

...

COMMUNICATION_FAILURE

Proper operation of the object is dependent on communication with a remote sensor or device and communication with the remote sensor or device has been lost.

TRIPPED

The actuator is not responding to commands, prevented by a tripped condition or by being mechanically held open.

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.

[Add new Clause 12.X, "Lighting Output Object Type", p. 251]

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
Lighting_Command	BACnetLightingCommand	W
In_Process	BOOLEAN	O
Description	CharacterString	O
Fade_Time	REAL	O
Ramp_Rate	REAL	O
Blink_Time	Unsigned	O ¹
Blink_Priority_Threshold	Unsigned	O ¹
Off_Delay	REAL	O ¹
On_Delay	REAL	O
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
Change_Of_State_Time	BACnetDateTime	O ²
Change_Of_State_Count	Unsigned	O ²
Time_Of_State_Count_Reset	BACnetDateTime	O ²
Elapsed_Active_Time	Unsigned32	O ³
Time_Of_Active_Time_Reset	BACnetDateTime	O ³
Min_Pres_Value	REAL	O ⁴
Max_Pres_Value	REAL	O ⁴
Lighting_Command_Priority	Unsigned	R
Power_On_Value	BACnetOptionalREAL	O
System_Failure_Value	BACnetOptionalREAL	O
Binary_Active_Value	REAL	O ⁵
Binary_Inactive_Value	REAL	O ⁵
Polarity	BACnetPolarity	O ⁵
Minimum_Off_Time	Unsigned32	O ⁶
Minimum_On_Time	Unsigned32	O ⁶
Profile_Name	CharacterString	O

¹ These properties are required if the object supports warning blink.

- ² 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.
- ³ 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.
- ⁴ If either of these properties is present, they shall both be present, and they are required to be writable.
- ⁵ If any of these properties are present, they shall all be present.
- ⁶ If either of these properties is present, they shall both be present.

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 `On_Delay` or `Off_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 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`. 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 class 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 seconds over which certain fade operations take place
duration	an overall time duration in seconds 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.

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, RAMP_UP, RAMP_DOWN, RAMP_UP_AT_RATE, RAMP_DOWN_AT_RATE, STEP_UP, STEP_DOWN, STEP_UP_BY, STEP_DOWN_BY, GOTO_LEVEL, 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	level is the target level. The calculated level is changed proportionally from a starting value until it reaches level over a period defined by the Fade_Time property.
FADE_TO_OVER	level is the target level. The calculated level is changed proportionally from its starting value until it reaches level over a period defined by fade-time .
RAMP_TO	level 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 level .
RAMP_TO_AT_RATE	level is the target level. The calculated level is changed by adding or subtracting an amount, proportional over time according to ramp-rate , to/from the starting level until it reaches level .
RAMP_UP	The target level is set to 100%. The calculated level is increased from the starting level until it reaches the target level by adding an amount, proportional over time according to the Ramp_Rate property.
RAMP_UP_AT_RATE	The target level is set to 100%. The calculated level is increased from the starting level until it reaches the target level by adding an amount, proportional over time according to ramp-rate .
RAMP_DOWN	The target level is set to 1%. The calculated level is decreased from the starting level until it reaches the target level by subtracting an amount, proportional over time according to the Ramp_Rate property.
RAMP_DOWN_AT_RATE	The target level is set to 1%. The calculated level is decreased from the starting level until it reaches the target level by subtracting an amount, proportional over time according to ramp-rate .
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 step-increment .
STEP_DOWN_BY	The calculated level is decreased from the starting level by subtracting an amount according to the step-increment .
GOTO_LEVEL	level 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.7 In_Process

This optional 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.8 Description

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

12.X.9 Fade_Time

This optional property, of type REAL, indicates the amount of time in seconds 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.

12.X.10 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.11 Blink_Time

This optional property, of type Unsigned, indicates the time in hundredths of seconds that the output shall blink off when Present_Value becomes a 0% value (a blink warning). A Blink_Time of zero shall disable this blink-off effect. If the Blink_Time property is present, then the Off_Delay and Blink_Priority_Threshold properties shall also be present.

12.X.11.1 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%, the output value is saved, and 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 Off_Delay seconds. When Blink_Time is zero and Present_Value becomes 0%, the lights shall remain on and not blink for up to Off_Delay seconds. Whether there is a single blink or multiple blinks shall be a local matter. If Off_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. If Present_Value is written at a priority lower numbered (more important than) than Blink_Priority_Threshold, then the blink shall not occur and the higher numbered (less important) priority slot shall not be relinquished.

12.X.12 Blink_Priority_Threshold

This optional property, of type Unsigned, indicates a write priority level of 1 to 16. After a blink warning with a non-zero Off_Delay, if the Present_Value had been written with a priority less important than Blink_Priority_Threshold, then the automatic relinquish of Present_Value shall occur. If the Blink_Priority_Threshold property is present, then the Blink_Time and Off_Delay properties shall also be present.

12.X.13 Off_Delay

This optional property, of type REAL, indicates the time in seconds after a blink warning until the Present_Value is automatically relinquished. An Off_Delay of zero disables this automatic relinquish. If the Off_Delay property is present, then the Blink_Time and Blink_Priority_Threshold properties shall also be present.

12.X.14 On_Delay

This optional property, of type REAL, indicates the time in seconds that the lights wait before turning on. When the Present_Value is changed from zero to some value greater than zero, the action is delayed for the time specified by this property.

12.X.15 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.16 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.17 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.18 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.19 Priority_Array

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

12.X.20 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.21 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.22 Change_Of_State_Time

This 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.23 Change_Of_State_Count

This 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.24 Time_Of_State_Count_Reset

This 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.25 Elapsed_Active_Time

This property, of type Unsigned32, represents the accumulated number of seconds that the Present_Value property has been ACTIVE since this property was most recently set to a zero value. ACTIVE shall be defined as the Binary_Active_Value property if it is present, or a non-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.26 Time_Of_Active_Time_Reset

This 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.27 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.28 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.29 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.30 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. Implementors may decide to restore the output to the Relinquish_Default value after a power failure or to restore the output to the value it had before the power was turned off or failed, the choice being a local matter. 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.31 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 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.32 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 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.

12.X.33 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.34 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.35 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
>= Binary_Active_Value	NORMAL	ON or ACTIVE
<= Binary_Inactive_Value	NORMAL	OFF or INACTIVE
>= Binary_Active_Value	REVERSE	OFF or INACTIVE
<= Binary_Inactive_Value	REVERSE	ON or ACTIVE

12.X.36 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.

12.X.37 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 a non-zero value if the Binary_Inactive_Value property is not present. The mechanism by which this is accomplished is a local matter.

[Add to 21, new BACnetLightingCommand production, p. 418]

-- 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] REAL OPTIONAL,
    duration                 [5] Unsigned OPTIONAL
}
```

[Add to 21, new BACnetLightingOperation production, p. 418]

```
BACnetLightingOperation ::= ENUMERATED {
    stop                      (0),
    fade-to                   (1),
    fade-to-over              (2),
    ramp-to                   (3),
    ramp-to-at-rate           (4),
    ramp-up                   (5),
    ramp-up-at-rate           (6),
    ramp-down                 (7),
    ramp-down-at-rate         (8),
    step-up                   (9),
    step-down                 (10),
    step-up-by                (11),
}
```

```

    step-down-by          (12),
    goto-level            (13),
    relinquish            (14)
}

```

[Add to **21**, new **BACnetOptionalReal** production, p. 422]

```

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

```

[Change **21**, **BACnetObjectType** enumeration, p. 421]

[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     (31),
    loop                (12),
    ...
    -- see lighting-output (31),
    ...
}

```

[Change **21**, **BACnetObjectTypesSupported** production, p. 422]

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

[Note: the order here is shown after an erratum was applied.]

```

BACnetObjectTypesSupported ::= BIT STRING {
    ...
    -- life-safety-zone    (22),
    -- lighting-output     (31),
    loop                  (12),
    ...
    pulse-converter       (24) (24),
    lighting-output       (31)
}

```

[Change **21**, **BACnetPropertyIdentifier** production , pp. 423-428]

[Note: other enumerations in the range 192 to 235 are used in Addenda *b-2*, *d-1*, *e-1*, and *f-1* to ANSI/ASHRAE 135-2004.]

```

BACnetPropertyIdentifier ::= ENUMERATED {
    ...
    bias                  (14),
    binary-active-value   (324),
    binary-inactive-value (325),
    blink-priority-threshold (216),
    blink-time            (217),
    buffer-size           (126),
    ...
    exception-schedule    (38),
    fade-time             (223),
    fault-values          (39),
    ...
    life-safety-alarm-values (166),
}

```

<i>lighting-command</i>	(224),
<i>lighting-command-priority</i>	(225),
limit-enable	(52),
...	
object-type	(79),
<i>off-delay</i>	(236),
<i>on-delay</i>	(237),
operation-expected	(161),
...	
polarity	(84),
<i>power</i>	(238),
<i>power-on-value</i>	(239),
prescale	(185),
...	
program-state	(92),
<i>progress-value</i>	(240),
proportional-constant	(93),
...	
protocol-version	(98),
<i>ramp-rate</i>	(241),
read-only	(99),
...	
status-flags	(111),
<i>step-increment</i>	(242),
stop-time	(143),
...	
stop-when-full	(144),
<i>system-failure-value</i>	(243),
system-status	(112),
...	
-- see value-change-time	(192),
...	
-- see <i>blink-priority-threshold</i>	(216),
-- see <i>blink-time</i>	(217),
...	
-- see <i>fade-time</i>	(223),
-- see <i>lighting-command</i>	(224),
-- see <i>lighting-command-priority</i>	(225),
...	
-- see <i>off-delay</i>	(236),
-- see <i>on-delay</i>	(237),
-- see <i>power</i>	(238),
-- see <i>power-on-value</i>	(239),
-- see <i>progress-value</i>	(240),
-- see <i>ramp-rate</i>	(241),
-- see <i>step-increment</i>	(242),
-- see <i>system-failure-value</i>	(243),
...	
-- see <i>binary-active-value</i>	(324),
-- see <i>binary-inactive-value</i>	(325),
}	

[Change **21**, **BACnetReliability** enumeration, p. 429]

[Note: enumerations 11 and 12 are used in Addendum 135-2004b-2.]

BACnetReliability ::= ENUMERATED {

```

...
communication-failure (12),
tripped (13),
...
}

```

```

-- 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.

```

[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 [240] REAL,
lighting-command [224] BACnetLightingCommand,
in-process [47] BOOLEAN OPTIONAL,
description [28] CharacterString OPTIONAL,
fade-time [223] REAL OPTIONAL,
ramp-rate [241] REAL OPTIONAL,
blink-time [217] Unsigned OPTIONAL,
off-delay [236] REAL OPTIONAL,
on-delay [237] REAL OPTIONAL,
blink-priority-threshold [216] Unsigned OPTIONAL,
step-increment [242] REAL OPTIONAL,
status-flags [111] BACnetStatusFlags,
reliability [103] BACnetReliability OPTIONAL,
out-of-service [81] BOOLEAN,
priority-array [87] BACnetPriorityArray,
relinquish-default [104] REAL,
power [238] 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 [225] Unsigned,
power-on-value [239] BACnetOptionalREAL OPTIONAL,
system-failure-value [243] BACnetOptionalREAL OPTIONAL,
profile-name [167] CharacterString OPTIONAL,
binary-active-value [324] REAL OPTIONAL,
binary-inactive-value [325] REAL OPTIONAL,
polarity [84] BACnetPolarity OPTIONAL,
minimum-off-time [66] Unsigned32 OPTIONAL,
minimum-on-time [67] Unsigned32 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 =	100
Property:	Ramp_Rate =	0.0
Property:	Blink_Time =	70
Property:	Off_Delay =	120
Property:	On_Delay =	0
Property:	Blink_Priority_Threshold =	9
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:	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:	Power_On_Value =	0.0
Property:	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

135-2004i-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 135-2004i-2

[Change **12.3.9**, p. 144]

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 **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}.