



**BSR/ASHRAE Addendum cx to
ANSI/ASHRAE Standard 135-2024**

Public Review Draft

**Proposed Addendum cx to Standard
135-2024, BACnet[®] - A Data
Communication Protocol for Building
Automation and Control Networks**

**First Publication Public Review (May 2026)
(Draft shows Proposed Changes to Current Standard)**

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

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

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

© 2026 ASHRAE. This draft is covered under ASHRAE copyright. Permission to reproduce or redistribute all or any part of this document must be obtained from the ASHRAE Manager of Standards, 180 Technology Parkway NW, Peachtree Corners, GA 30092. Phone: 404-636-8400, Ext. 1125. Fax: 404-321-5478. E-mail: standards.section@ashrae.org.

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

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

FOREWORD

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

135-2024*cx*-1 Group Command Object Type, p. 3

135-2024*cx*-2 Lighting Status Object Type, p. 11

135-2024*cx*-3 Clarify definitions of luminance and range as they apply to lighting control, p. 18

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2024 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 at this time. All other material in this document is provided for context only and is not open for public review comment except as it relates to the proposed changes.

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

135-2024cx-1 Group Command Object Type

Rationale

The WriteGroup service was added to the BACnet standard and was designed for servers to execute in their Channel objects, but there is no standard object to enable configuration of the data used in the WriteGroup service initiation. This change makes this service configurable over the network.

[add to 12, Group Command Object Type, pg.163]

12.X Group Command Object Type

The Group Command object type defines a standardized object whose properties configure and initiate group-based write operations using the WriteGroup service. It allows for coordinated control of multiple properties across multiple objects, potentially spanning multiple devices.

The Group Command object is ideal for creating and activating lighting scenes. A lighting scene is a predefined configuration of light levels or lighting commands for one or more channels of lights that are members of a Channel object. For example, a "presentation" scene in a conference room might dim the main lights and turn on accent lighting. The Channel object and Lighting Output object also support a BACnetLightingCommand datatype which can be used to activate scenes that ramp up, ramp down, or stop the ramp of a Channel object that includes a Lighting Output object member.

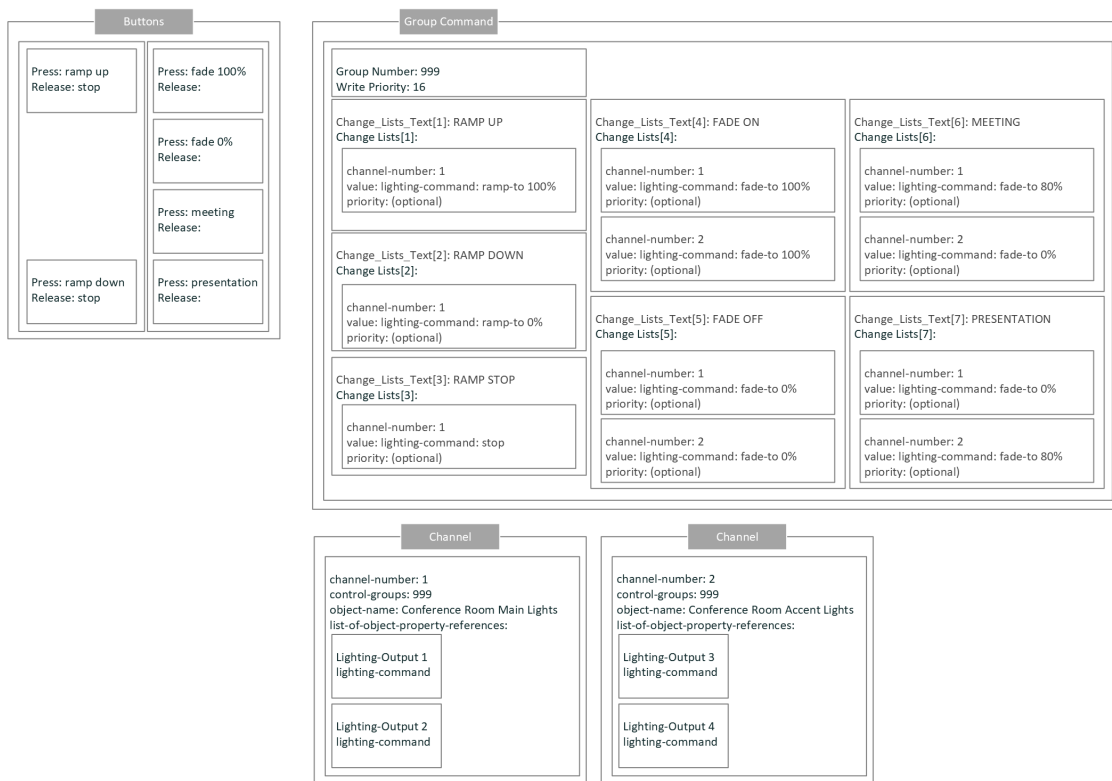


Figure 12-X1. Example use of the Group Command Object Type for lighting scenes

Group Command objects may optionally support intrinsic reporting to facilitate the reporting of fault conditions. Group Command objects that are capable of detecting faults shall only perform the first stage of reliability-evaluation and shall not apply a standardized fault algorithm. Group Command objects that support intrinsic reporting shall not apply an event algorithm.

The Group Command object and its properties are summarized in Table 12-X1 and described in detail in this clause.

Table 12-X1. Properties of the Group Command Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	Unsigned	W
Group_Number	Unsigned32	R
Priority_For_Writing	Unsigned(1..16)	R
Inhibit_Delay	Boolean	R
Change_Lists	BACnetARRAY[N] of BACnetGroupChannelValues	R
Change_Lists_Text	BACnetARRAY[N] of CharacterString	O
Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R
Status_Flags	BACnetStatusFlags	O ¹
Event_State	BACnetEventState	O ¹
Reliability	BACnetReliability	O ¹
Event_Detection_Enable	Boolean	O ^{1,2}
Notification_Class	Unsigned	O ^{1,2}
Event_Enable	BACnetEventTransitionBits	O ^{1,2}
Acked_Transitions	BACnetEventTransitionBits	O ^{1,2}
Notify_Type	BACnetNotifyType	O ^{1,2}
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	O ^{1,2}
Event_Message_Texts	BACnetARRAY[3] of CharacterString	O ²
Event_Message_Texts_Config	BACnetARRAY[3] of CharacterString	O ²
Reliability_Evaluation_Inhibit	Boolean	O ³
Value_Source	BACnetValueSource	O ^{4,5,6}
Audit_Level	BACnetAuditLevel	O ⁷
Auditable_Operations	BACnetAuditOperationFlags	O ⁷
Tags	BACnetARRAY[N] of BACnetNameValue	O
Profile_Location	CharacterString	O
Profile_Name	CharacterString	O

- ¹ These properties are required if the object supports intrinsic reporting.
- ² These properties shall be present only if the object supports intrinsic reporting.
- ³ If this property is present, then the Reliability property shall be present.
- ⁴ This property is required if the object supports the value source mechanism.
- ⁵ This property shall be present only if the object supports the value source mechanism.
- ⁶ This property shall be writable as described in Clause 19.5.
- ⁷ This property shall be present only if the device supports audit 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 read-only property, of type BACnetObjectType, indicates membership in a particular object type class. The value of this property shall be GROUP_COMMAND.

12.X.4 Description

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

12.X.5 Present_Value

This property, of type Unsigned, indicates which change the Group Command object is to write or has already written. Whenever the Present_Value property is written to, it triggers the Group Command object to initiate a WriteGroup service request.

The Present_Value may be written to with any value from 0 to the maximum number of changes supported by the Change_Lists property. When the Present_Value is written to, the Group Command object initiates a WriteGroup-Request service request. The new value of the Present_Value property determines which list of changes the Group Command object shall write. These changes are specified in the Change_Lists property, which is an array of lists of changes to write. The array is indexed by the value being written. A given list may be empty, in which case no changes are written. If the list is not empty and the Present_Value is written, then a WriteGroup service request is initiated with the WriteGroup service primitives specified by this Group Command object and the specific change list index of the Present_Value property, as shown in Table 12-X2.

12.X.6 Group_Number

This property, of type Unsigned32, shall specify the 'Group Number' primitive used to construct the WriteGroup service request. A Group_Number of zero shall mean "no assignment."

12.X.7 Priority_For_Writing

This property, of type Unsigned, shall specify the 'Write Priority' primitive used to construct the WriteGroup service request.

12.X.8 Inhibit_Delay

This property, of type Boolean, shall specify the 'Inhibit Delay' primitive used to construct the WriteGroup service request.

12.X.9 Change_Lists

This property, of type BACnetARRAY of BACnetGroupChannelValues, specifies an array of "change lists." These change lists are indexed by the value that is written to the Present_Value property. A given list may be empty, in which case no action takes place. The value zero is a special case that takes no action and behaves like an empty list. The number of defined change lists may be found by reading the Change_Lists property with an array index of zero. If the size of this array is changed, the size of the Change_Lists_Text array, if present, shall also be changed to the same size.

If the list is not empty and the Group_Number is not zero and the Present_Value is written, then a WriteGroup service request is initiated with the WriteGroup service primitives specified by Group Command object properties and the specific change list index of the Present_Value property, as shown in Table 12-X2.

Table 12-X2. WriteGroup Service Primitives specified by Group Command Object Properties

Parameter Name	Property Identifier
Argument	
Group Number	Group_Number
Write Priority	Priority_For_Writing
Change List	Change_Lists[Present_Value]
Inhibit Delay	Inhibit_Delay

Each BACnetGroupChannelValues is a specification of the Change List service primitive used in the WriteGroup service request.

12.X.9.1 Initializing New Array Elements When the Array Size is Increased

If the size of this array is increased by writing to array index zero, each new array element shall contain an empty list.

12.X.9.2 Decreasing the Array Size

If the size of the array is decreased by writing to array index zero with a value less than Present_Value, the Present_Value shall be set to zero.

12.X.9.3 Changes to the Array Element Values

If an array value is changed by writing a different value, the Present_Value shall be set to zero.

12.X.10 Change_Lists_Text

This property, of type BACnetARRAY of CharacterString, shall be used to indicate a text string description for each of the possible values of the Present_Value property. The content of these strings is not restricted. If the size of this array is changed, the size of the Change_Lists array shall also be changed to the same size.

12.X.11 Property_List

This read-only property, of type BACnetARRAY of BACnetPropertyIdentifier, contains one BACnetPropertyIdentifier 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.12 Status_Flags

This read-only property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of the Group Command object. One flag is 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 TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise 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 The value of this flag shall be logical FALSE (0).

OUT_OF_SERVICE The value of this flag shall be logical FALSE (0).

12.X.13 Event_State

This read-only 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.14 Reliability

This property, of type BACnetReliability, provides an indication of whether the Group Command object is properly configured and is able to process write requests received. Failing property write attempts initiated by this object shall not cause this property to indicate a fault condition. Such error conditions shall be indicated in respective other properties.

12.X.15 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.16 Notification_Class

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

12.X.17 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.18 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 13.2.3). Each flag shall have the value TRUE if no event of that type has ever occurred for the object.

12.X.19 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.20 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 BACnetDateTime shall have X'FF' in each octet, and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

12.X.21 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, respectively (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.22 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.23 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.

12.X.24 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.25 Audit_Level

This property, of type BACnetAuditLevel, specifies the level of auditing to perform for the specific object. If this property has the value DEFAULT, or if this property is not present in the object, then the audit level value for the object shall be taken from the Audit_Level property of the object's associated Audit Reporter object.

For details on auditing and the use of this property, see Clause 19.6.3.

If this property is present and not configurable, the value shall not be NONE.

12.X.26 Auditable_Operations

This property, of type BACnetAuditOperationFlags, specifies the operations that the device will report for this object.

If present and not configurable, the values of the READ, NOTIFICATION, and SUBSCRIPTION bits shall be 0, and the WRITE, CREATE, DELETE, and ACKNOWLEDGE_ALARM bits shall be 1.

If this property is not present, and the device supports auditing, then the Auditable_Operations property of the object's associated Audit Reporter object shall control the operations that are auditable for this object.

For details on auditing and the use of this property, see Clause 19.6.3.

12.X.27 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.28 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.29 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.

[Add to Clause 21, page 1492]

```
BACnetGroupChannelValues ::= SEQUENCE {  
    values [0] SEQUENCE OF BACnetGroupChannelValue  
}
```

[Add to Clause 21, BACnetObjectType]

Group Command (n),

[Add to Clause 21, BACnetObjectTypesSupported]

Group Command (n),

[Add to Clause 21, BACnetPropertyIdentifier, add in alphabetical order]

...
change-lists (n),
change-lists-text (n+1),
...
group-number (n+2),
...
inhibit-delay (n+3),
...
-- see *change-lists* (n),
-- see *change-lists-text* (n+1),
-- see *group-number* (n+2),
-- see *inhibit-delay* (n+3),
...

135-2024cx-2 Lighting Status Object Type

Rationale

When using Lighting Outputs and Binary Lighting Output to represent a group of luminaire statuses, it would be beneficial to collect the status to get aggregated metrics such as power, actual ranges and so on. It has been noticed that modelling a lighting system with Lighting Outputs controlling subordinate Lighting Outputs results in issues with controlling priorities.

[Add to Clause 3 in alphabetic order]

local broadcast: a message addressed to all devices or nodes on the same BACnet network as the originator.

luminaire: *a representation of a physical light-emitting entity, it may also be referred to as ballast, led-driver, or control-gear.*

medium: the physical transmission entity. Typical media are twisted-pair wire, fiber optic cable, and coaxial cable.

[Add new Clause 12.x,]

12.X Lighting Status Object Type

The Lighting Status object type defines a standardized object whose properties represent the status and characteristics of a luminaire. It is intended to be used together with or without a controlling BACnet object (e.g. Lighting Output object) as shown in Figures 12-x1 and Figure 12-x2.

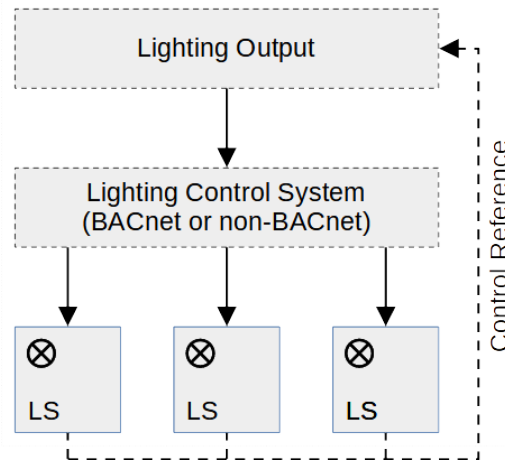


Figure 12-x1. An example showing a Lighting Output object controlling three Lighting Status objects through an intermediary Lighting Control System.

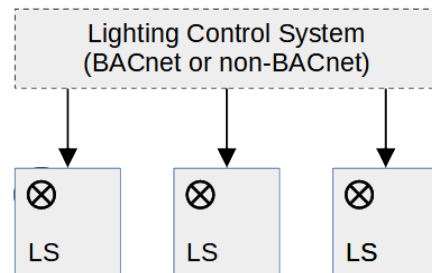


Figure 12-x2. An example showing three Lighting Status objects through a Lighting Control System.

The object and its properties are summarized in Table 12-X1 and described in detail in this subclause.

Table 12-X1. Properties of the Lighting Status Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	Real	R
Description	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Reliability	BACnetReliability	O
Out_Of_Service	Boolean	R
Units	BACnetEngineeringUnits	R
Control_Reference	BACnetDeviceObjectReference	R
Power	Real	O

Instantaneous_Power	Real	O
Color_Temperature	Unsigned	O
Color	BACnetxyColor	O
Property_List	BACnetARRAY[N] of BACnetPropertyIdentifier	R
Audit_Level	BACnetAuditLevel	O ^{1,2}
Auditable_Operations	BACnetAuditOperationFlags	O ^{1,2}
Tags	BACnetARRAY[N] of BACnetNameValue	O
Profile_Location	CharacterString	O
Profile_Name	CharacterString	O

¹ If either of these properties is present, they shall both be present

² This property shall be present only if the device supports audit 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 read-only property, of type BACnetObjectType, indicates membership in a particular object type class. The value of this property shall be LIGHTING_STATUS.

12.X.4 Present_Value

This property, of type Real, indicates the lighting level of the luminaire, in percent. The frequency at which this property is updated is a local matter. For analog luminaires, the valid range shall be 0 and 1 to 100% where 0% means off, and 1%(dimkest) to 100%(brightest). For binary luminaires, the value 0% means off and all other values mean on.

The Present_Value property shall be writable when Out_Of_Service is TRUE.

12.X.5 Description

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

12.X.6 Status_Flags

This read-only property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a Lighting Status 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).

OUT_OF_SERVICE Logical TRUE (1) if the Out_Of_Service property has a value of TRUE, otherwise logical FALSE (0).

12.X.7 Reliability

This property, of type BACnetReliability, provides an indication of whether the Present_Value or the operation of the luminaire in question is "reliable" as far as the BACnet device or operator can determine and, if not, why. This property shall be writable when Out_Of_Service is TRUE.

12.X.8 Out_Of_Service

The Out_Of_Service property, of type Boolean, is an indication whether (TRUE) or not (FALSE) the luminaire that the object represents is not in service.

When Out_Of_Service is TRUE:

- (a) the Present_Value property is decoupled from the luminaire and will not track changes to the luminaire
- (b) the Reliability property, if present, and the corresponding state of the FAULT flag of the Status_Flags property shall be decoupled from the luminaire
- (c) the Present_Value property and the Reliability property, if present and capable of taking on values other than NO_FAULT_DETECTED, shall be writable to allow simulating specific conditions or for testing purposes.
- (d) 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 in the luminaire.

12.X.9 Units

This read-only property, of type BACnetEngineeringUnits, indicates the measurement units of this object. Its value shall be PERCENT.

12.X.10 Control_Reference

This property, of type BACnetDeviceObjectReference, is a reference to the object controlling the luminaire. If this luminaire is not controlled by a BACnet object, then it shall be uninitialized.

This property shall be considered uninitialized if it contains 4194303 in the instance part of the object identifier and in the device instance part of the device identifier if present.

12.X.11 Power

This property, of type Real, is the nominal power, in kilowatts, of the load(s) controlled by this object when the luminaire level is 100% (brightest).

12.X.12 Instantaneous_Power

This read-only property, of type Real, is the current nominal power, in kilowatts, of the luminaire represented by this object.

12.X.13 Color_Temperature

This read-only property, of type Unsigned, is the current color temperature of this luminaire, in Kelvin, in the range 1000 .. 30000. The value 0 shall indicate that the color-temperature is unknown.

12.X.14 Color

This read-only property, of type BACnetxyColor, is the current color of this luminaire. The range of values for x and y shall be 0.0 to 1.0. The value (0,0) shall indicate that the color is unknown.

12.X.15 Property_List

This read-only property, of type BACnetARRAY of BACnetPropertyIdentifier, contains one BACnetPropertyIdentifier 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.16 Audit Level

This property, of type BACnetAuditLevel, specifies the level of auditing to perform for the specific object. If this property has the value DEFAULT, or if this property is not present in the object, then the audit level value for the object shall be taken from the Audit_Level property of the object's associated Audit Reporter object.

For details on auditing and the use of this property, see Clause 19.6.3.

If this property is present and not configurable, the value shall not be NONE.

12.X.17 Auditable_Operations

This property, of type BACnetAuditOperationFlags, specifies the operations that the device will report for this object.

If present and not configurable, the values of the READ, NOTIFICATION, and SUBSCRIPTION bits shall be 0, and the WRITE, CREATE, DELETE, and ACKNOWLEDGE_ALARM bits shall be 1.

If this property is not present, and the device supports auditing, then the Auditable_Operations property of the object's associated Audit Reporter object shall control the operations that are auditable for this object.

For details on auditing and the use of this property, see Clause 19.6.3.

12.X.18 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.19 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.20 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.

[Insert into Table 12.64]

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

Property Identifier	Property Datatype	Conformance Code
...
Command_Time_Array	BACnetARRAY[16] of BACnetTimeStamp	O ⁷
Group_Members	BACnetARRAY[N] of BACnetDeviceObjectReference	O
Units	BACnetEngineeringUnits	R

[Add a new property after the Command_Time_Array]

12.54.34 Command_Time_Array

This read-only property, of type BACnetARRAY[16] of BACnetTimeStamp, indicates the time at which each priority was last commanded or relinquished. See Clause 19.5 for a description of the value source mechanism.

12.54.X1 Group_Members

This property, of type BACnetARRAY of BACnetDeviceObjectReference, holds an array of references to BACnet objects which represent objects that are associated with the luminaires that this object controls such as Lighting_Status. It is a local matter as to how this array is used and which objects are referenced in this array.

12.54.X2 Units

This read-only property, of type BACnetEngineeringUnits, indicates the measurement units of this object. Its value shall be PERCENT.

[Insert into Table 12.69]

Table 12-69. Properties of the Binary Lighting Output Object Type

Property Identifier	Property Datatype	Conformance Code
...
Command_Time_Array	BACnetARRAY[16] of BACnetTimeStamp	O ⁷
Group_Members	BACnetARRAY[N] of BACnetDeviceObjectReference	O
Inactive_Text	CharacterString	O ^x
Active_Text	CharacterString	O ^x
...

^x *If one of the optional properties Inactive_Text or Active_Text is present, then both of these properties shall be present.*

[Add a new property after the Command_Time_Array]

12.55.36 Command_Time_Array

This read-only property, of type BACnetARRAY[16] of BACnetTimeStamp, indicates the time at which each priority was last commanded or relinquished. See Clause 19.5 for a description of the value source mechanism.

12.55.X1 Group_Members

This property, of type BACnetARRAY of BACnetDeviceObjectReference, holds an array of references to BACnet objects which represent objects that are associated with the luminaires that this object controls such as Lighting_Status. It is a local matter as to how this array is used and which objects are referenced in this array.

12.55.X2 Inactive_Text

This property, of type CharacterString, characterizes the intended effect of the INACTIVE state of the Present_Value property. The content of this string is a local matter, but it is intended to represent a human-readable description of the INACTIVE state. If either the Active_Text or the Inactive_Text is present, then both properties shall be present.

12.55.X3 Active_Text

This property, of type CharacterString, characterizes the intended effect of the ACTIVE state of the Present_Value property. The content of this string is a local matter, but it is intended to represent a human-readable description of the ACTIVE state. If either the Active_Text or the Inactive_Text is present, then both properties shall be present.

[Add to **Clause 3.2**, p. 16]

3.2 Terms Defined for this Standard

...

luminance (lighting control): *The brightness of a lighting output's physical output level. See non-normalized range.*

...

non-normalized range (lighting control): *The physical output level, or non-normalized range, is specified as the linearized percentage (0..100%) of the possible light output range ("luminance") with 0.0% being off, 1.0% being dimmest, and 100.0% being brightest.*

normalized range (lighting control): *A range of values from 0.0% to 100.0% representing lighting luminance. The value 0.0% shall signify "off" or no luminance. Values from 1.0% to 100.0% shall represent a "linearized" sub-range of the lighting luminance from the `Min_Actual_Value` to the `Max_Actual_Value` which themselves represent values in the "non-normalized range" of the actual output luminance from 0 to 100%. See 12.54.*

...

operating range (lighting control): *A range of values representing lighting luminance within the normalized range that represent the acceptable values for control as defined by the `High_End_Trim` and `Low_End_Trim`. See 12.54.*

[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	X	Addendum <i>cx</i> to ANSI/ASHRAE Standard 135-2024 Approved by ASHRAE on MONTH DAY, 20XX; and by the American National Standards Institute on MONTH DAY, 20XX. <ol style="list-style-type: none">1. Addition of Group Command object type2. Addition of Lighting Status object type3. Clarify definitions of luminance and range in lighting control