



**BSI Standards Publication**

## **Building automation and control systems (BACS)**

---

Part 5: Data communication protocol

## National foreword

This British Standard is the UK implementation of EN ISO 16484-5:2017+A1:2020. It is identical to ISO 16484-5:2017, incorporating amendment 1:2020. It supersedes BS EN ISO 16484-5:2017, which is withdrawn.

The text of ISO amendment 1:2020 has been provided in its entirety at the beginning of this document. BSI's policy of providing consolidated content remains unchanged; however, in the interest of expediency, in this instance BSI have chosen to collate the relevant content at the beginning of this document.

The UK participation in its preparation was entrusted to Technical Committee RHE/2, Ventilation for buildings, heating and hot water services.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2020  
Published by BSI Standards Limited 2020

ISBN 978 0 539 06862 7

ICS 35.240.67; 91.040.01

**Compliance with a British Standard cannot confer immunity from legal obligations.**

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 March 2018.

### Amendments/corrigenda issued since publication

| Date        | Text affected                                                       |
|-------------|---------------------------------------------------------------------|
| 31 May 2020 | Implementation of ISO amendment 1:2020 with CEN endorsement A1:2020 |

English Version

## Building automation and control systems (BACS) - Part 5: Data communication protocol (ISO 16484-5:2017)

Systèmes d'automatisation et de gestion technique du  
bâtiment - Partie 5: Protocole de communication de  
données (ISO 16484-5:2017)

Systeme der Gebäudeautomation - Teil 5:  
Datenkommunikationsprotokoll (ISO 16484-5:2017)

This European Standard was approved by CEN on 18 July 2017.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION  
COMITÉ EUROPÉEN DE NORMALISATION  
EUROPÄISCHES KOMITEE FÜR NORMUNG

**CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels**

## **European foreword**

This document (EN ISO 16484-5:2017) has been prepared by Technical Committee ISO/TC 205 "Building environment design" in collaboration with Technical Committee CEN/TC 247 "Building Automation, Controls and Building Management" the secretariat of which is held by SNV.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by January 2018 and conflicting national standards shall be withdrawn at the latest by January 2018.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN shall not be held responsible for identifying any or all such patent rights.

This document supersedes EN ISO 16484-5:2014.

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

### **Endorsement notice**

The text of ISO 16484-5:2017 has been approved by CEN as EN ISO 16484-5:2017 without any modification.

## **Foreword to amendment A1**

This document (EN ISO 16484-5:2017/A1:2020) has been prepared by Technical Committee ISO/TC 205 "Building environment design" in collaboration with Technical Committee CEN/TC 247 "Building Automation, Controls and Building Management" the secretariat of which is held by SNV.

This Amendment to the European Standard EN ISO 16484-5:2017 shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by November 2020, and conflicting national standards shall be withdrawn at the latest by November 2020.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN shall not be held responsible for identifying any or all such patent rights.

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.

### **Endorsement notice**

The text of ISO 16484-5:2017/Amd 1:2020 has been approved by CEN as EN ISO 16484-5:2017/A1:2020 without any modification.



**Documents**

Document Sharing Hub

---

---

**Building automation and control  
systems (BACS) —**

**Part 5:  
Data communication protocol  
AMENDMENT 1**

*Systèmes d'automatisation et de gestion technique du bâtiment —  
Partie 5: Protocole de communication de données  
AMENDEMENT 1*





## **COPYRIGHT PROTECTED DOCUMENT**

© ISO 2020

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
CP 401 • Ch. de Blandonnet 8  
CH-1214 Vernier, Geneva  
Phone: +41 22 749 01 11  
Fax: +41 22 749 09 47  
Email: [copyright@iso.org](mailto:copyright@iso.org)  
Website: [www.iso.org](http://www.iso.org)

Published in Switzerland

## Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. International Standards are drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](http://www.iso.org/directives)).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see [www.iso.org/patents](http://www.iso.org/patents)).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see [www.iso.org/iso/foreword.html](http://www.iso.org/iso/foreword.html).

This document was prepared by Technical Committee ISO/TC 205, *Building environmental design*, in collaboration with the European Committee for Standardization (CEN) Technical Committee CEN/TC 247, *Building Automation, Controls and Building Management*, in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement).

A list of all parts in the ISO 16484 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](http://www.iso.org/members.html).

(This foreword is not part of the standard. It is merely informative and does not contain requirements necessary for conformance to the standard.)

## FOREWORD

The purpose of this addendum is to add several independent substantive changes to the BACnet standard. The changes are summarized below.

- 135-2016bd-1. Add Staging Object Type.
- 135-2016be-1. Add Lighting BIBBS and Device Profiles.
- 135-2016bi -1. Add Audit Reporting.
- 135-2016bi -2. Change DeviceCommunicationControlService for Audit Reporting.
- 135-2016bi -3. Modify Logging Objects to Allow for Extremely Large Logs.
- 135-2016bk -1. Expand the reserved range of BACnetPropertyIdentifier.
- 135-2016bl - 1. Clarify Result(-) response for failed WritePropertyMultiple requests.
- 135-2016bl - 2. Clarify ReadPropertyMultiple response on OPTIONAL when empty.
- 135-2016bl - 3. Clarify Out Of Service.
- 135-2016bm-1. Reduce allowed range for Usage Timeout.
- 135-2016bm-2. Specify design choices for MS/TP devices.
- 125-2016bm-3. Handle unwanted MS/TP frames in IDLE state.
- 135-2016bn-1. Make SCHED BIBBs consistent on supported datatypes, and add BOOLEAN.
- 135-2016bn-2. Clarify COV and COVP related BIBBs.
- 135-2016bn-3. Clock is required for support of AE-ACK-A.
- 135-2016bp-1. Make rules for POST consistent with rules for PUT.
- 135-2016bp-2. Make 'type' consistent at all levels and introduce 'effectiveType'.
- 135-2016bp-3. Fully specify the behavior of "includes".
- 135-2016bp-4. Remove the path syntax from the 'select' query parameter.
- 135-2016bp-5. Resolve conflicting statements about configuring external authorization servers.
- 135-2016bp-6. Remove incorrect table for callback formats.
- 135-2016bp-7. Allow plain text POSTs for primitive data.
- 135-2016bp-8. Allow extended error numbers.
- 135-2016bp-9. Add new error numbers.
- 135-2016bp-10. Add formal definition for JSON equivalent to XML's <CSML>.
- 135-2016bp-11. Specify 'name' safety check for setting data.
- 135-2016bp-12. Specify how to evaluate relative paths for collections of links.
- 135-2016bp-13. Allow proprietary categories for the 'metadata' query.
- 135-2016bq-1. Fix the Absentee\_Limit property of the Access Credential object type.
- 135-2016bq-2. Ensure that the denied or granted access event is generated last.

In the following document, language to be added to existing clauses of EN ISO 16484-5 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.

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

# Building automation and control systems (BACS) —

## Part 5:

## Data communication protocol

### AMENDMENT 1

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

##### Rationale

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

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

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

[Insert new **Clause 12.X**]

#### 12.X Staging Object Type

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

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

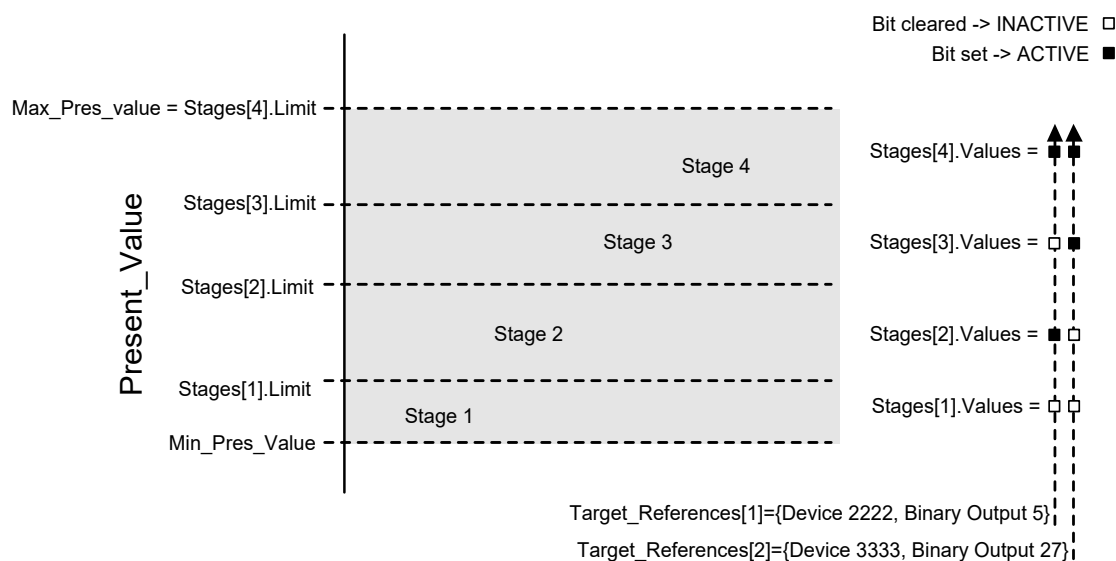


**Figure 12-X.** Typical Staging Application ( $N_{\text{stages}}=4$ ,  $N_{\text{references}}=2$ )

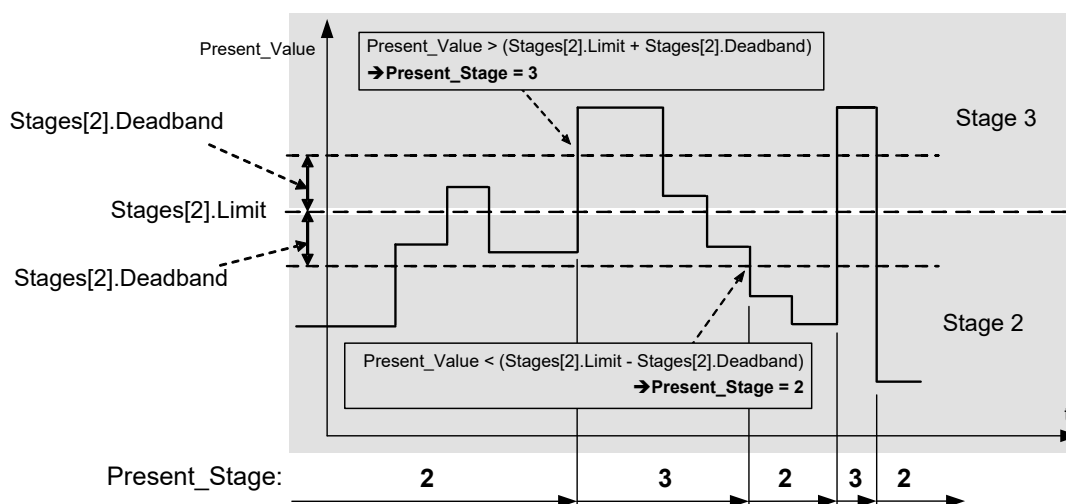


**Figure 12-X2.** Stage Limits incorporate hysteresis through the use of a Deadband around each Limit

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

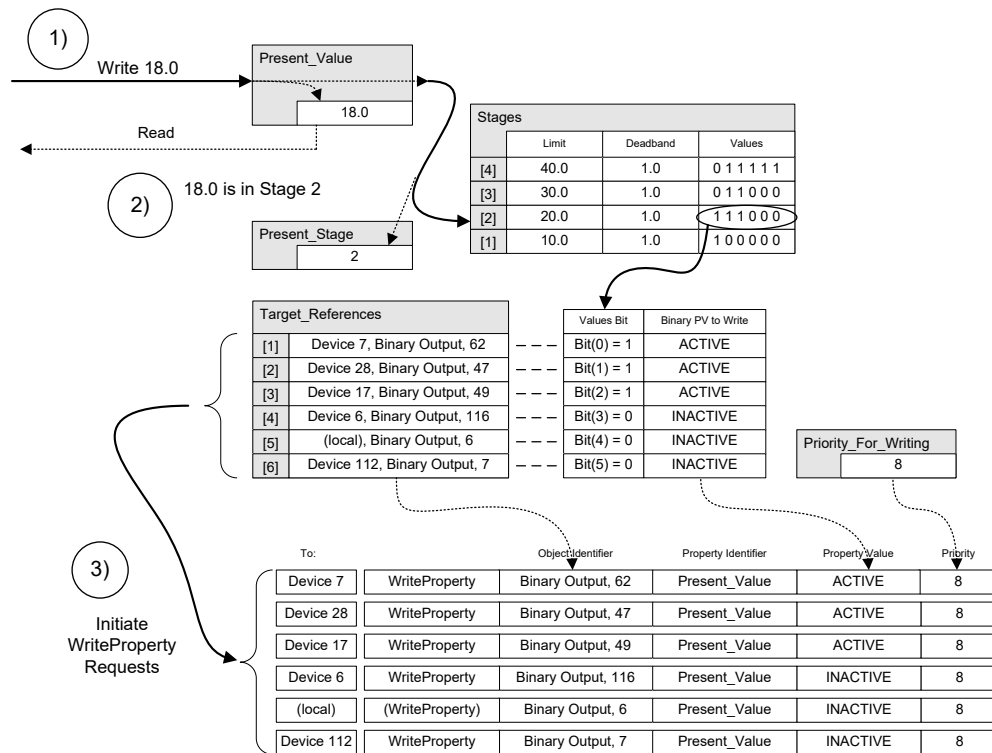


**Figure 12-X3.** Pipeline of operations when Present\_Value is written

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

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

**Table 12-X.** Properties of the Staging Object Type

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

<sup>1</sup> The array size of this property is N<sub>stages</sub>.

<sup>2</sup> The array size of this property is N<sub>references</sub>.

<sup>3</sup> This property is required if the object supports COV reporting.

<sup>4</sup> These properties are required if the object supports intrinsic reporting.

<sup>5</sup> These properties shall be present only if the object supports intrinsic reporting.

<sup>6</sup> This property, if present, is required to be read-only.

<sup>7</sup> This property is required if the object supports the value source mechanism.

<sup>8</sup> This property shall be present only if the object supports the value source mechanism.

<sup>9</sup> This property shall be writable as described in Clause 19.5.

### 12.X.1 Object\_Identifier

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

### 12.X.2 Object\_Name

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

### 12.X.3 Object\_Type

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

### 12.X.4 Present\_Value

This property, of type REAL, indicates the current value, in engineering units, of the Staging object. If Present\_Value is written with a value less than Min\_Pres\_Value, then it shall be clamped to Min\_Pres\_Value. If Present\_Value is written with a value greater than Max\_Pres\_Value, then it shall be clamped to Max\_Pres\_Value.

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

```
ops = current Present_Stage
npv = new Present_Value

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

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

Present_Value = npv
```

**Figure 12-X4.** Pseudocode Algorithm for Evaluating Present\_Value and Present\_Stage

#### 12.X.4.1 Writing to Referenced Objects

Changes to Present\_Stage shall cause a write of ACTIVE or INACTIVE to the corresponding object's Present\_Value for each Target\_References array element. For each bit (Index = 0 to N<sub>references</sub> -1) in the Stages[Present\_Stage].Values bitstring, if the bit is set (1), an ACTIVE value shall be written, or if clear (0), then an INACTIVE value shall be written to the Target\_References[Index +1] object's Present\_Value.

Writes to Present\_Value that subsequently trigger writing to referenced objects due to reevaluation of Present\_Stage, are not expected to wait until the reference writes occur before returning a Result(+) or Result(-) for the write to Present\_Value. Subsequently if any write to a referenced object fails, Reliability shall be changed to COMMUNICATION\_FAILURE. The COMMUNICATION\_FAILURE shall remain in effect until all reference writes have been completed successfully. How a particular implementation handles other failures during writing to referenced objects shall be a local matter except that Reliability shall indicate a value other than NO\_FAULT\_DETECTED.

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

### 12.X.5 Present\_Stage

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

Attempts to read Present\_Stage when it is internally set to 0 shall return a Result(-) with an 'Error Class' of PROPERTY and an 'Error Code' of VALUE\_NOT\_INITIALIZED.

### 12.X.6 Stages

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

|          |            |
|----------|------------|
| Limit    | REAL       |
| Values   | BIT STRING |
| Deadband | REAL       |

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

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

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

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

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

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

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

If Reliability has the value CONFIGURATION\_ERROR, then Present\_Value shall be set to Min\_Pres\_Value and Present\_Stage to 1.

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

If the Stages property is written, the value of Present\_Stage shall be reevaluated and corresponding writes of new values to Target\_References' objects shall be triggered.

If the size of this array is changed, the size of the Stage\_Names array shall also be changed to the same size.

### 12.X.7 Stage\_Names

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

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

### 12.X.8 Status\_Flags

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

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

where:

|                |                                                                                                                       |
|----------------|-----------------------------------------------------------------------------------------------------------------------|
| IN_ALARM       | Logical FALSE (0) if the Event_State property has a value of NORMAL, otherwise logical TRUE (1).                      |
| FAULT          | Logical TRUE (1) if the Reliability property does not have a value of NO_FAULT_DETECTED, otherwise logical FALSE (0). |
| OVERRIDDEN     | Always logical FALSE (0).                                                                                             |
| OUT_OF_SERVICE | Logical TRUE (1) if the Out_Of_Service property has a value of TRUE, otherwise logical FALSE(0).                      |

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

### 12.X.9 Event\_State

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

### 12.X.10 Reliability

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

Table 12-X2 summarizes scenarios when Reliability has a value other than NO\_FAULT\_DETECTED.

**Table 12-X2.** Reliability Scenarios

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

### 12.X.11 Out\_Of\_Service

The Out\_Of\_Service property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present\_Value property is controllable by software local to the BACnet device.

When Out\_Of\_Service is TRUE:

- changes to the Present\_Value property are decoupled from the Target\_References. This means that the objects referenced by Target\_References shall not be updated;

- b) the Present\_Value property and the Reliability property, if capable of taking on values other than NO\_FAULT\_DETECTED, shall be writable to allow simulating specific conditions or for testing purposes;
- c) other functions that depend on the state of the Present\_Value or Reliability properties shall respond to changes made to these properties, as if those changes had occurred while the object was in service;

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

When Out\_Of\_Service becomes FALSE, it shall trigger updating of referenced objects as described in Clause 12.X.4.1.

#### 12.X.12 Description

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

#### 12.X.13 Units

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

#### 12.X.14 Target\_References

This property, of type BACnetARRAY[N] of BACnetDeviceObjectReference, is an array representing references to Binary Output, Binary Value, or Binary Lighting Output objects. The size of the array is N<sub>references</sub>. The Target\_References array elements [Index = 1 to N<sub>references</sub>] shall correspond to the bit(Index - 1) of all 'Values' bitstrings of the Stages property.

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

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

#### 12.X.15 Priority\_For\_Writing

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

#### 12.X.16 Default\_Present\_Value

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

#### 12.X.17 Min\_Pres\_Value

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

If a change to Min\_Pres\_Value, Stages[1].Limit, or Stages[1].Deadband violates this rule, then the Reliability property shall have a value of CONFIGURATION\_ERROR. If a change to Min\_Pres\_Value causes it to be greater than Present\_Value, then Present\_Value shall be written with Min\_Pres\_Value.

#### 12.X.18 Max\_Pres\_Value

This read-only property, of type REAL, represents the maximum value for Present\_Value. This value shall be defined as the 'Limit' of the last entry of the Stages property array (i.e., Stages[N<sub>stages</sub>].Limit).

If the size of the Stages array is zero, then Max\_Pres\_Value shall be equal to Min\_Pres\_Value.

#### 12.X.19 COV\_Increment

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

**12.X.20 Notification\_Class**

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

**12.X.21 Event\_Enable**

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

**12.X.22 Acked\_Transitions**

This read-only property, of type BACnetEventTransitionBits, shall convey three flags that separately indicate the acknowledgment state for TO\_OFFNORMAL, TO\_FAULT, and TO\_NORMAL events (see Clause 13.2.2.1.5). Each flag shall have the value TRUE if no event of that type has ever occurred for the object.

**12.X.23 Notify\_Type**

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

**12.X.24 Event\_Time\_Stamps**

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

**12.X.25 Event\_Message\_Texts**

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

**12.X.26 Event\_Message\_Texts\_Config**

This property, of type BACnetARRAY[3] of CharacterString, contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO\_OFFNORMAL, TO\_FAULT, and TO\_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

**12.X.27 Event\_Detection\_Enable**

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

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

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

**12.X.28 Reliability\_Evaluation\_Inhibit**

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

When reliability-evaluation is disabled, the Reliability property shall have the value NO\_FAULT\_DETECTED unless Out\_Of\_Service is TRUE and an alternate value has been written to the Reliability property.

**12.X.29 Property\_List**

This read-only property is a BACnetARRAY of property identifiers, one property identifier for each property that exists within the object. The Object\_Name, Object\_Type, Object\_Identifier, and Property\_List properties are not included in the list.

**12.X.30 Value\_Source**

This property, of type BACnetValueSource, indicates the source of the value of the Present\_Value. The Value\_Source property and its use in the value source mechanism are described in Clause 19.5.

**12.X.31 Tags**

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

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

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

**12.X.32 Profile\_Location**

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

If a Profile\_Location value is not provided for a particular object, then the client shall use the Profile\_Location of the Device object, if provided, to find the definition of the Profile\_Name.

**12.X.33 Profile\_Name**

This property, of type CharacterString, is the name of an object profile to which this object conforms. To ensure uniqueness, a profile name shall begin with a vendor identifier code (see Clause 23) in base-10 integer format, followed by a dash. All subsequent characters are administered by the organization registered with that vendor identifier code. The vendor identifier code that prefixes the profile name shall indicate the organization that publishes and maintains the profile. This vendor identifier need not have any relationship to the vendor identifier of the device within which the object resides.

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

[Change Table 13-1]

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

| Object Type    | Criteria                                                                                                                        | Properties Reported                                   |
|----------------|---------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------|
| ...            |                                                                                                                                 |                                                       |
| <i>Staging</i> | <i>If Present_Value changes by COV_Increment<br/>or<br/>Status_Flags changes at all<br/>or<br/>Present_Stage changes at all</i> | <i>Present_Value, Status_Flags,<br/>Present_Stage</i> |

[Change Table 13-5]

**Table 13-5.** Properties Reported in CHANGE\_OF\_RELIABILITY Notifications

| Object Type    | Properties                             |
|----------------|----------------------------------------|
| ...            | ...                                    |
| <i>Staging</i> | <i>Present_Value<br/>Present_Stage</i> |
| Timer          | ...                                    |

1 ...

[Change **Clause 21**, **BACnetObjectType** and **BACnetObjectTypesSupported** productions]

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

```

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

```

**BACnetObjectTypesSupported** ::= BIT STRING {

```

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

```

[Change **Clause 21**, **BACnetPropertyIdentifier** production]

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

```

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

```

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

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

[Insert into **Table K-1** in **DS-V-A**]

| <b><i>Staging</i></b> |
|-----------------------|
| <i>Object_Name</i>    |
| <i>Present_Value</i>  |
| <i>Present_Stage</i>  |
| <i>Status_Flags</i>   |
| <i>Units</i>          |

[Insert into **Table K-5** in **DS-M-A**]

| <b><i>Staging</i></b> |
|-----------------------|
| <i>Present_Value</i>  |
| <i>Out Of Service</i> |

[Change **Clause K.1.27**]

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

...

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

...

[Change **Clause K.1.28**]

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

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

...

[Change Clause **K.1.29**]

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

...

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

...

[Change Clause **K.1.30**]

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

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

...

[Change Clause **K.1.31**]

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

...

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

...

[Change Clause **K.1.32**]

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

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

...

[Change Clause K.1.33]

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

...

Devices claiming conformance to this BIBB shall be capable of commanding and relinquishing standard commandable properties at priority 8 (other priorities may also be supported) of those objects listed in Table K-5 excluding Averaging, Loop, Accumulator, Pulse Converter, Channel, Lighting Output, ~~and Binary Lighting Output objects~~, *and Staging objects*, and writing the properties listed in Table K-5 and Table K-10, excluding Averaging, Loop, Accumulator, Pulse Converter, Channel, Lighting Output, ~~and Binary Lighting Output objects~~, *and Staging objects*.

[Change Clause K.1.34]

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

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

**135-2016be-1 Add Lighting BIBBs and Device Profiles****Rationale**

With the addition of the Lighting Output and Binary Lighting Output object types, there is a need for lighting specific BIBBs and device profiles.

[Add new Clauses K.1.X1 to K.1.X11]

**K.1.X1 BIBB – Data Sharing-Lighting Output -A (DS-LO-A)**

The A device modifies properties in Lighting Output, Binary Lighting Output, Analog Output, Analog Value, Binary Output, Binary Value, Multi-State Output, and Multi-State Value object types.

| BACnet Service | Initiate | Execute |
|----------------|----------|---------|
| WriteProperty  | x        |         |

Devices claiming conformance to DS-LO-A shall be able to write to the Present\_Value property of the Lighting Output, Binary Lighting Output, Analog Output, Analog Value, Binary Output, Binary Value, Multi-State Output, and Multi-State Value object types.

A device claiming support for DS-LO-A is interoperable with devices that support any combination of DS-LO-B, DS-BLO-B, or DS-WP-B in the case of Analog Output, Analog Value, Binary Output, Binary Value, Multi-State Output, and Multi-State Value object types.

**K.1.X2 BIBB – Data Sharing-Lighting Output Status-A (DS-LOS-A)**

The A device reads properties in Lighting Output, Binary Lighting Output, Analog Output, Analog Value, Binary Output, Binary Value, Multi-State Output, and Multi-State Value object types.

| BACnet Service | Initiate | Execute |
|----------------|----------|---------|
| ReadProperty   | x        |         |

Devices claiming conformance to DS-LOS-A shall be able to read the Present\_Value and Egress\_Active properties of Lighting Output and Binary Lighting Output object types and to read the Present\_Value property of Analog Output, Analog Value, Binary Output, Binary Value, Multi-State Output, and Multi-State Value object types.

A device claiming support for DS-LOS-A is interoperable with devices that support any combination of DS-LO-B, DS-BLO-B, or DS-RP-B in the case of Analog Output, Analog Value, Binary Output, Binary Value, Multi-State Output, and Multi-State Value object types.

#### K.1.X3 BIBB – Data Sharing-Advanced Lighting Output-A (DS-ALO-A)

The A device modifies properties in Lighting Output, Binary Lighting Output, Analog Output, Analog Value, Binary Output, Binary Value, Multi-State Output, and Multi-State Value object types.

| BACnet Service | Initiate | Execute |
|----------------|----------|---------|
| WriteProperty  | x        |         |

The A device shall be capable of modifying all of the standard properties of the Lighting Output, Binary Lighting Output, Channel, Analog Output, Analog Value, Binary Output, Binary Value, Multi-State Output, and Multi-State Value object types.

A device claiming support for DS-ALO-A is interoperable with devices that support any combination of DS-LO-B, DS-BLO-B, DS-RP-B, or DS-WP-B in the case of Analog Output, Analog Value, Binary Output, Binary Value, Multi-State Output, and Multi-State Value object types.

#### K.1.X4 BIBB - Data Sharing-Lighting Output-B (DS-LO-B)

The B device implements the Lighting Output object type.

| BACnet Service | Initiate | Execute |
|----------------|----------|---------|
| ReadProperty   |          | x       |
| WriteProperty  |          | x       |

Devices claiming conformance to DS-LO-B shall support the Lighting Output object type.

A device claiming support for DS-LO-B is interoperable with devices that support DS-LO-A or DS-ALO-A.

#### K.1.X5 BIBB - Data Sharing-Binary Lighting Output-B (DS-BLO-B)

The B device implements the Binary Lighting Output object type.

| BACnet Service | Initiate | Execute |
|----------------|----------|---------|
| ReadProperty   |          | x       |
| WriteProperty  |          | x       |

Devices claiming conformance to DS-BLO-B shall support the Binary Lighting Output object type.

A device claiming support for DS-BLO-B is interoperable with devices that support DS-LO-A or DS-ALO-A.

#### K.1.X6 BIBB – Device Management – Lighting Output Management-A (DM-LOM-A)

BACnet allows lighting object instances to be dynamically created and deleted. The A device shall be able to dynamically create and delete Lighting Output, Binary Lighting Output, and Channel object types supported by the B device.

| BACnet Service | Initiate | Execute |
|----------------|----------|---------|
| CreateObject   | x        |         |
| DeleteObject   | x        |         |

**K.1.X7 BIBB - Data Sharing-Lighting View-A (DS-LV-A)**

The A device retrieves values from a minimum set of lighting objects and properties and presents them to the user. Devices claiming conformance to this BIBB shall support DS-RP-A. The A device shall be capable of using ReadProperty to retrieve any of the properties listed below. The A device may use alternate services where support for execution of the alternate service is supported by the B device.

| BACnet Service | Initiate | Execute |
|----------------|----------|---------|
| ReadProperty   | x        |         |

Devices claiming conformance to this BIBB shall be capable of reading and displaying the object properties listed in Table K-X1.

**Table K-X1. Object Properties for Which Presentation Is Required**

| Lighting Output                                                                                                                     | Binary Lighting Output                                                                        | Analog Output, Analog Value, Binary Output, Binary Value, Multi-State Output, Multi-State Value | Channel                                      |
|-------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|----------------------------------------------|
| Object_Name<br>Present_Value<br>Tracking_Value<br>Lighting_Command<br>Status_Flags<br>Blink_Warn_Enable<br>Current_Command_Priority | Object_Name<br>Present_Value<br>Status_Flags<br>Blink_Warn_Enable<br>Current_Command_Priority | Object_Name<br>Present_Value<br>Status_Flags                                                    | Object_Name<br>Present_Value<br>Status_Flags |

The format of a presented property value is unrestricted; the intent of this BIBB is not to impose how, or in what form, a device displays data values. For example, enumerated values could be displayed as icons, references could be displayed using the referenced object's name, and numerical values could be displayed graphically.

Actions taken by the A device when retrieval of a value for display fails are a local matter.

Devices claiming conformance to this BIBB are not required to support presentation of objects and properties that are introduced in a Protocol\_Revision newer than that claimed by the A device.

A device claiming support for this BIBB is interoperable with devices that support DS-RP-B and one or more of the objects listed in Table K-X1.

**K.1.X8 BIBB - Data Sharing-Lighting Advanced View-A (DS-LAV-A)**

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

| BACnet Service | Initiate | Execute |
|----------------|----------|---------|
| ReadProperty   | x        |         |

The information conveyed by the properties in Table K-2 can be otherwise determined and as such need not be read and presented by devices claiming conformance to this BIBB.

In order to ensure that products that claim support for this BIBB are capable of presenting accurate data values across the full range of values for each data type, devices claiming support for this BIBB shall be able to meet the requirements described in Table K-3. In addition, the device shall be able to present all valid values for fields of the Lighting\_Command property. The format is unrestricted as long as each valid value is distinguishable.

For Character String property values, the A device shall be capable of presenting string values for specific BACnet properties with at least the number of characters, independent of their encoding, specified in Table K-4.

The above presentation requirements are not required to be applied in all circumstances, but rather shall be available for every property value in the system. This should allow a product to restrict its presentation under specific conditions yet still allow the user full access to any specific property value.

The A device shall be capable of reading and presenting all standard forms of the datatypes as defined per the A device's claimed Protocol\_Revision.

Actions taken by the A device when retrieval of a value for display fails are a local matter.

Devices claiming conformance to this BIBB are not required to support presentation of objects and properties that are introduced in a Protocol\_Revision newer than that claimed by the A device.

A device claiming support for this BIBB is interoperable with devices that support DS-RP-B and one or more of the objects listed in Table K-1.

#### K.1.X9 BIBB – Data Sharing-Lighting Modify-A (DS-LM-A)

The A device writes properties of standard objects that are generally expected to be adjusted during normal operation of the lighting system. Devices claiming support for this BIBB are not expected to be capable of fully configuring lighting controller BACnet devices, although they are not inherently restricted from doing so.

| BACnet Service | Initiate | Execute |
|----------------|----------|---------|
| WriteProperty  | x        |         |

Devices claiming conformance to this BIBB shall be capable of commanding and relinquishing standard commandable properties at priority 8 (other priorities may also be supported) of those objects listed in Table K-X3 and writing the properties listed in Table K-X3.

**Table K-X3.** Standard Properties That DS-LM-A Devices Shall Be Capable of Writing

| Lighting Output                   | Binary Lighting Output | Analog Output, Analog Value, Binary Output, Binary Value, Multi-State Output, Multi-State Value | Channel       |
|-----------------------------------|------------------------|-------------------------------------------------------------------------------------------------|---------------|
| Present_Value<br>Lighting_Command | Present_Value          | Present_Value                                                                                   | Present_Value |

Devices claiming support for this BIBB shall be capable of writing values within the full range as defined in Table K-6. In addition, the device shall be able to manipulate all of the fields of the Lighting\_Command property.

Devices claiming conformance to this BIBB are not required to support presentation and modification of objects and properties that are introduced in a Protocol\_Revision newer than that claimed by the A device.

A device claiming support for this BIBB is interoperable with devices that support DS-WP-B and support one or more of the objects listed in Table K-X3.

#### K.1.X10 BIBB – Data Sharing-Lighting Advanced Modify-A (DS-LAM-A)

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

| BACnet Service | Initiate | Execute |
|----------------|----------|---------|
| WriteProperty  | x        |         |

Devices claiming support for this BIBB shall be capable of writing values within the full range as defined in Table K-6. In addition, the device shall be able to manipulate all of the fields of the Lighting\_Command property.

The A device shall be capable of writing all standard forms of the datatypes as defined per the A device's claimed Protocol\_Revision.

Devices claiming conformance to this BIBB are not required to support presentation and modification of objects and properties that are introduced in a Protocol\_Revision newer than that claimed by the A device.

A device claiming support for this BIBB is interoperable with devices that support DS-WP-B and support one or more of the objects listed in Table K-5.

[Change **Annex A**]

[Add new device profile checkboxes for lighting devices]

#### **BACnet Standardized Device Profiles (Annex L):**

- ☐ BACnet Cross-Domain Advanced Operator Workstation (B-XAWS)
- ☐ BACnet Advanced Operator Workstation (B-AWS)
- ☐ BACnet Operator Workstation (B-OWS)
- ☐ BACnet Operator Display (B-OD)
- ☐ *BACnet Advanced Lighting Workstation (B-ALWS)*
- ☐ *BACnet Lighting Operator Display (B-LOD)*
- ☐ BACnet Advanced Life Safety Workstation (B-ALSWS)
- ...
- ☐ BACnet Access Control Security Display (B-ACSD)
- ☐ *BACnet Advanced Lighting Control Station (B-ALCS)*
- ☐ *BACnet Lighting Control Station (B-LCS)*
- ☐ BACnet Building Controller (B-BC)
- ...
- ☐ BACnet Smart Sensor (B-SS)
- ☐ *BACnet Lighting Supervisor (B-LS)*
- ☐ *BACnet Lighting Device (B-LD)*
- ☐ BACnet Advanced Life Safety Controller (B-ALSC)
- ...

[Change **Annex L**]

#### **ANNEX L – DESCRIPTIONS AND PROFILES OF STANDARDIZED BACnet DEVICES (NORMATIVE)**

(This annex is part of this standard and is required for its use.)

This annex provides descriptions of “standardized” types of BACnet devices. Any device that implements all the required BACnet capabilities for a particular device type and interoperability area may claim to be a device of that particular type. Devices may also provide additional capabilities and shall indicate these capabilities in their PICS.

BACnet device profiles are categorized into families:

- Operator Interfaces. This family is composed of B-XAWS, B-AWS, B-OWS, and B-OD.
- *Lighting Operator Interfaces. This family is composed of B-XAWS, B-ALWS, and B-LOD.*
- Life Safety Operator Interfaces. This family is composed of B-ALSWS, B-LSWS, and B-LSAP.
- Access Control Operator Interfaces. This family is composed of B-XAWS, B-AACWS, B-ACWS, and B-ACSD.
- *Lighting Control Stations. This family is composed of B-ALCS and B-LCS.*
- Controllers. This family is composed of B-BC, B-AAC, B-ASC, B-SA, and B-SS.

- *Lighting Controllers.* This family is composed of B-LS and B-LD.
- *Life Safety Controllers.* This family is composed of B-ALSC, and B-LSC.
- *Access Control Controllers.* This family is composed of B-AACC, and B-ACC.
- *Miscellaneous.* This family is composed of B-RTR, B-GW, B-BBMD, B-ACDC, and B-ACCR.

...

[Change **Clause L.1.1**]**L.1.1 BACnet Cross-Domain Advanced Workstation (B-XAWS)**

The B-XAWS workstation is an advanced operator workstation for all building automation domains except life safety that includes the functionality of the following device profiles:

- B-AWS, see Clause L.1.2
- B-AACWS, see Clause L.3.1
- B-ALWS, see Clause L.X.1

[Insert new **Clause L.X**]**L.X Lighting Operator Interface Profiles**

The following table indicates which BIBBs shall be supported by the device types of this family, for each interoperability area. The B-XAWS is not shown in this table. See Clause L.1.1.

| Data Sharing |           | Alarm & Event Management |       |
|--------------|-----------|--------------------------|-------|
| B-ALWS       | B-LOD     | B-ALWS                   | B-LOD |
| DS-RP-A,B    | DS-RP-A,B | AE-N-A                   |       |
| DS-RPM-A     |           | AE-ACK-A                 |       |
| DS-WP-A      | DS-WP-A   | AE-AS-A                  |       |
| DS-WPM-A     |           | AE-AVM-A                 |       |
| DS-LAV-A     | DS-LV-A   | AE-AVN-A                 |       |
| DS-LAM-A     | DS-LM-A   | AE-ELVM-A                |       |
| DS-WG-A      | DS-WG-A   |                          |       |
| DS-ALO-A     | DS-ALO-A  |                          |       |

| Scheduling  |       | Trending |       |
|-------------|-------|----------|-------|
| B-ALWS      | B-LOD | B-ALWS   | B-LOD |
| SCHED-AVM-A |       | T-AVM-A  |       |

| Device & Network Management |            |
|-----------------------------|------------|
| B-ALWS                      | B-LOD      |
| DM-DDB-A,B                  | DM-DDB-A,B |
| DM-ANM-A                    |            |
| DM-ADM-A                    |            |
| DM-DOB-B                    | DM-DOB-B   |
| DM-DCC-A                    |            |
| DM-MTS-A                    |            |
| DM-OCD-A                    |            |
| DM-RD-A                     |            |
| DM-BR-A                     |            |

### L.X.1 BACnet Advanced Lighting Workstation (B-ALWS)

The B-ALWS workstation is an advanced operator workstation with abilities to monitor, control, and configure lighting devices.

The B-ALWS profile enables the specification of the following:

#### Data Sharing

- Presentation of data (i.e., reports and graphics)
- Ability to monitor the value of BACnet objects relevant for lighting, including all required and optional properties
- Ability to modify lighting parameters
- Ability to create, delete, and configure Lighting Output, Binary Lighting Output, and Channel objects
- Ability to command lighting devices using WriteGroup
- Ability to configure advanced lighting parameters in lighting devices

#### Alarm and Event Management

- Operator notification and presentation of event information
- Alarm acknowledgment by operators
- Alarm summarization
- Adjustment of alarm limits and conditions
- Adjustment of alarm routing
- Ability to create, delete, and configure Event Enrollment, Notification Class, and Notification Forwarder objects
- Presentation and modification of Event Logs

#### Scheduling

- Modification of calendars and schedules
- Display of the start and stop times (schedule) of scheduled devices
- Display of calendars
- Creation and deletion of calendars and schedules

#### Trending

- Modification of the parameters of a Trend Log object
- Display of trend data
- Creation and deletion of new Trend Log objects

#### Device and Network Management

- Ability to find other BACnet devices
- Ability to find all objects in BACnet devices
- Ability to silence a device on the network that is transmitting erroneous data
- Ability to synchronize the time in devices across the BACnet internetwork at the request of the operator
- Ability to cause a remote device to reinitialize itself
- Ability to backup and restore the configuration of other devices

### L.X.2 BACnet Lighting Operator Display (B-LOD)

The B-LOD is a basic operator interface with limited capabilities relative to a B-ALWS. The B-LOD profile could be used for wall-mounted display devices, simple web server gateways to BACnet lighting devices, displays affixed to BACnet devices; handheld terminals or other very simple control stations.

The B-LOD profile enables the specification of the following:

#### Data Sharing

- Presentation of basic data
- Ability to monitor the value of BACnet objects relevant for lighting
- Ability to modify lighting parameters
- Ability to command lighting devices using WriteGroup
- Ability to configure advanced lighting parameters in lighting devices

#### Alarm and Event Management

- No requirement

#### Scheduling

- No requirement

## Trending

- No requirement

## Device and Network Management

- Ability to find other BACnet devices
- Ability to choose among discovered BACnet devices
- Ability to find objects in other BACnet devices
- Ability to choose among discovered BACnet device's objects

[Insert new **Clause L.Y**]

**L.Y Lighting Control Station Profiles**

The following table indicates which BIBBs shall be supported by the device types of this family, for each interoperability area.

| Data Sharing  |              | Alarm & Event Management |              |
|---------------|--------------|--------------------------|--------------|
| <b>B-ALCS</b> | <b>B-LCS</b> | <b>B-ALCS</b>            | <b>B-LCS</b> |
| DS-RP-A,B     | DS-RP-B      |                          |              |
| DS-RPM-A      |              |                          |              |
| DS-WP-A       | DS-WP-A      |                          |              |
| DS-WPM-A      |              |                          |              |
|               |              |                          |              |
| DS-WG-A       |              |                          |              |
| DS-ALO-A      | DS-LO-A      |                          |              |

| Scheduling    |              | Trending      |              |
|---------------|--------------|---------------|--------------|
| <b>B-ALCS</b> | <b>B-LCS</b> | <b>B-ALCS</b> | <b>B-LCS</b> |
| SCHED-E-B     |              |               |              |

| Device & Network Management |              |
|-----------------------------|--------------|
| <b>B-ALCS</b>               | <b>B-LCS</b> |
| DM-DDB-A,B                  | DM-DDB-A,B   |
| DM-DOB-B                    | DM-DOB-B     |
| DM-DCC-B                    | DM-DCC-B     |
| DM-TS-B<br>or DM-UTC-B      |              |

**L.Y.1 BACnet Advanced Lighting Control Station (B-ALCS)**

A B-ALCS is any device that needs to interact with Lighting Output, Binary Lighting Output, and Channel objects in other devices for the purpose of monitoring and/or controlling them. B-ALCS devices are not Workstations and Operator Displays. Instead, they are more limited devices nonetheless used for configuration. Examples might include smart wall switches/dimmers, scene preset consoles, etc. B-ALCS devices differ from B-LCS devices in B-ALCS's ability to configure and command advanced lighting parameters in lighting devices.

## Data Sharing

- Ability to monitor and command BACnet lighting devices
- Ability to provide the values of any of its BACnet objects
- Ability to command lighting devices using WriteGroup

## Alarm and Event Management

- No requirement

## Scheduling

- Ability to schedule output actions, both in the local device and in other devices, both binary and analog, based on date and time

## Trending

- No requirement

Device and Network Management

- Ability to find other BACnet devices
- Ability to respond to queries about its status
- Ability to respond to requests for information about any of its objects
- Ability to respond to communication control messages

**L.Y.2 BACnet Lighting Control Station (B-LCS)**

A B-LCS is any device that needs to interact with Lighting Output or Binary Lighting Output objects in other devices for the purpose of monitoring and/or controlling them. B-LCS devices are different from Workstations and Operator Displays in that their interactions with humans are more limited. Examples might include smart wall switches/dimmers, scene preset consoles, etc.

Data Sharing

- Ability to command BACnet lighting objects' Present\_Value properties
- Ability to provide the values of any of its BACnet objects

Alarm and Event Management

- No requirement

Scheduling

- No requirement

Trending

- No requirement

Device and Network Management

- Ability to find other BACnet devices
- Ability to respond to queries about its status
- Ability to respond to requests for information about any of its objects
- Ability to respond to communication control messages

[Insert new **Clause L.Z**]

**L.Z Lighting Controller Profiles**

The following table indicates which BIBBs shall be supported by the device types of this family, for each interoperability area.

| Data Sharing |                           | Alarm & Event Management |      |
|--------------|---------------------------|--------------------------|------|
| B-LS         | B-LD                      | B-LS                     | B-LD |
| DS-RP-B      | DS-RP-B                   |                          |      |
| DS-WP-A,B    | DS-WP-B                   |                          |      |
| DS-WG-E-B    |                           |                          |      |
| DS-ALO-A     | DS-BLO-B<br>or<br>DS-LO-B |                          |      |

| Scheduling |      | Trending |      |
|------------|------|----------|------|
| B-LS       | B-LD | B-LS     | B-LD |
| SCHED-E-B  |      |          |      |

| Device & Network Management |          |
|-----------------------------|----------|
| B-LS                        | B-LD     |
| DM-DDB-A,B                  | DM-DDB-B |
| DM-DOB-B                    | DM-DOB-B |
| DM-DCC-B                    | DM-DCC-B |
| DM-TS-B<br>or<br>DM-UTC-B   |          |

**L.Z.1 BACnet Lighting Supervisor (B-LS)**

A B-LS is any device that implements Channel objects and optionally Lighting Output and/or Binary Lighting Output objects with the ability to forward channel writes to other BACnet devices.

**Data Sharing**

- Ability to provide values for any of its BACnet objects upon request
- Ability to allow modification of some or all of its BACnet objects by another device
- Ability to execute WriteGroup commands
- Ability to propagate Channel values to objects external to the device

**Alarm and Event Management**

- No requirement

**Scheduling**

- Ability to schedule output actions, both in the local device and in other devices, both binary and analog, based on date and time

**Trending**

- No requirement

**Device and Network Management**

- Ability to find other BACnet devices
- Ability to respond to queries about its status
- Ability to respond to requests for information about any of its objects
- Ability to respond to communication control messages
- Ability to synchronize its internal clock upon request

**L.Z.2 BACnet Lighting Device (B-LD)**

A B-LD is any device that implements Binary Lighting Output and/or Lighting Output objects

**Data Sharing**

- Ability to provide values for any of its BACnet objects upon request
- Ability to allow modification of some or all of its BACnet objects by another device

**Alarm and Event Management**

- No requirement

**Scheduling**

- No requirement

**Trending**

- No requirement

**Device and Network Management**

- Ability to respond to queries about its status
- Ability to respond to requests for information about any of its objects
- Ability to respond to communication control messages

**135-2016bi-1. Add Audit Reporting.****Rationale**

The standard currently has no definitions for an interoperable controller and workstation based audit reporting and logging.

This addendum adds a new Audit Reporter object type and new audit notification services to report auditable actions. A new Audit Log object type and a new audit query service is added to log and retrieve audit notifications.

Both BACnet clients and servers are allowed to report auditable actions. Servers report changes to local objects, clients report successful and attempted changes along with extra information such as reason for change. The consumer of the logs will be responsible for correlating the multiple entries for a single action.

Configuration of audit reporting is supported on a per object basis with different levels of auditing.

[Insert new **Clause 19.Y**]

## 19.Y Audit Logging

Audit logging consists of recording records in audit logs which document the sequence of activities that affect the system.

BACnet audit logging is made up of four actors: the client that requests the operation being logged (the operation source), the server that performs the operation (the operation target), the logger that stores the audit records, and the viewer that consumes the audit log. In this framework, both the operation source and the operation target are given the opportunity to provide audit log records to the audit logger. This allows BACnet devices to provide these features:

- 1) audit logging when either the operation source or the operation target does not support audit logging;
- 2) logging of extended audit information that is not available for reporting by the operation target;
- 3) recording of failed actions by the operation source when the operation target does not respond;
- 4) recording of failed actions by the operation target where unauthorized operation sources do not log their attempts.

The operation source generates audit log entries which describe the operation it requested and which optionally contain extra information including the BACnet object which is the source of the operation (if applicable), user identification (if applicable), and a description of, or reason for, the operation.

The server generates audit log entries which describe the operation it has been requested to perform.

The logger receives notifications from local and remote operation sources and operation targets and places the notifications in Audit Log objects.

Operation source side auditing is controlled via:

- 1) an Audit Reporter object in the operation source device which generates the audit notifications; and
- 2) the Audit\_Notification\_Recipient property in the operation source's Device object.

Operation target side auditing is controlled via:

- 1) configuration properties in the object operated upon by the operation being reported;
- 2) Audit Reporter objects which generate the audit notifications; and
- 3) the Audit\_Notification\_Recipient property in the operation target's Device object.

The following figures provide a diagrammatic overview of the architecture discussed in the rest of this clause.

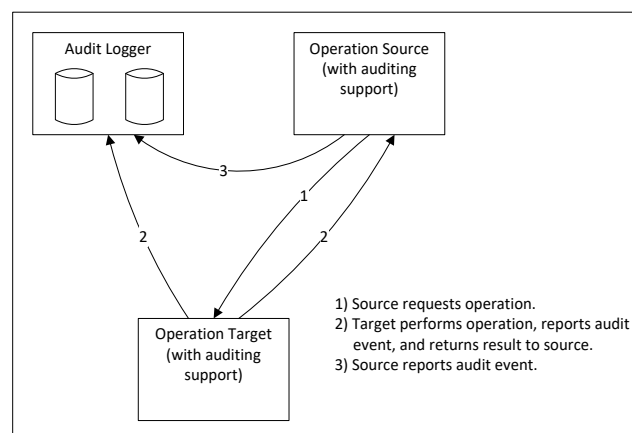


**Figure 19-Y1:** General flow of operations and audit notifications from both operation sources and operation targets.

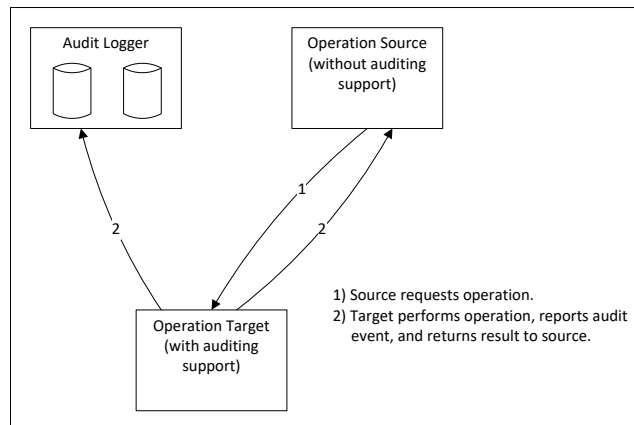


**Figure 19-Y2:** General flow of operations and audit notifications from operation targets with no operation source notifications.

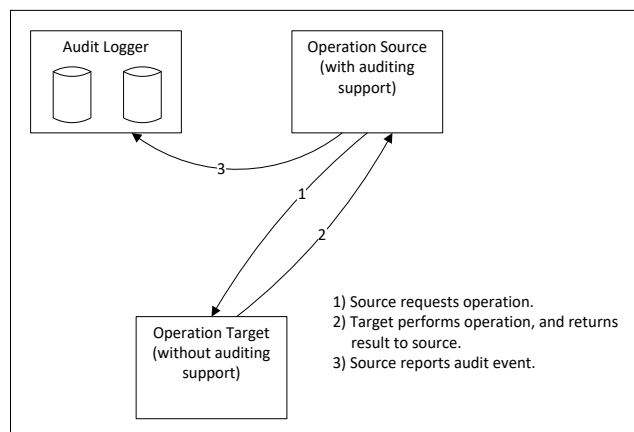


**Figure 19-Y3:** General flow of operations and audit notifications from operation source only.

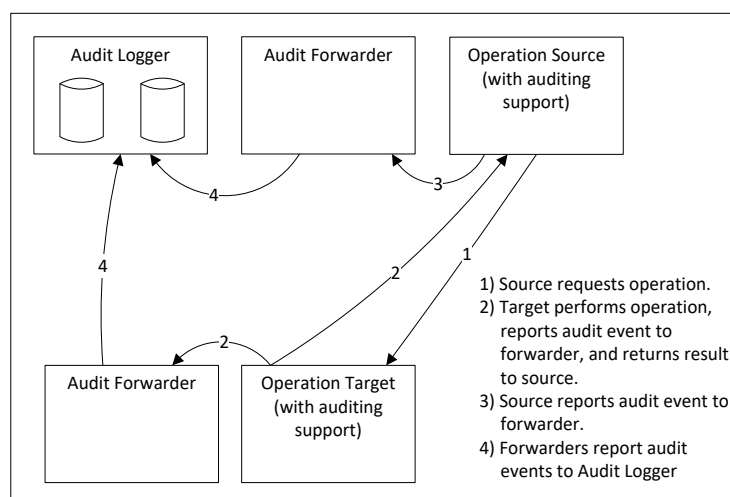


**Figure 19-Y4:** General flow of operations and audit notifications from both operation sources and operation targets with audit forwarders.

### 19.Y.1 Audit Notification Generation

Operation targets generate audit notifications for the following operations (subject to filtering):

- 1) a BACnet request for an operation type described in Table 19-Y3, which is executed successfully;
- 2) a local request for an operation type described in Table 19-Y3, which is successfully performed but which is not the result of the normal operation of the object (for example, updating of an input object's Present\_Value based on reading of the physical input would not be auditable, but the setting of the Present\_Value of a value object by a local program would be auditable); and
- 3) any action the local device determines to be relevant in an audit log.

It is a local matter whether or not an operation target generates audit notifications for BACnet requests for an operation type described in Table 19-Y3, which fail. It is important to consider when such operations will result in notifications as this could form the basis of a denial of service attack through the increased workload and network traffic required to report the failed operations.

Operation Sources generate audit notifications for the following operations (subject to filtering):

- 1) a BACnet request for an operation type described in Table 19-Y3, which is requested; and
- 2) any action the local device determines to be relevant in an audit log.

When reporting a successful operation, no result code is included in the audit notification. When reporting a failed operation, a result code shall be included.

#### 19.Y.1.1 Audit Notification Generation Through Monitoring

It is possible for a device that is not involved in the actions to monitor the network traffic and generate audit notifications on behalf of operation sources and/or operation targets.

Such a product is installed so that it is able to monitor all communications of interest that originate with an operation source (or all operation sources), or all communications destined for an operation target (or all operation targets). When BACnet messages are detected which contain auditable actions, the product generates audit records as if it were the operation source or the operation target device (depending on which device it is performing auditing for.)

The configuration and control of products that provide such functionality is outside the scope of this standard.

### 19.Y.2 Audit Reporter Objects

Audit notifications are generated by Audit Reporter objects. Both operation source and operation target devices which generate audit notifications contain Audit Reporter objects.

In the most basic case, there will be a single Audit Reporter object responsible for all audit notifications. In complex scenarios, a device may have multiple Audit Reporter objects each generating audit notifications for a different set of objects to allow for a finer granularity of control over audit notifications.

When a device supports audit reporting and is a BACnet client, exactly one Audit Reporter object shall have the Audit\_Source\_Reporter property set to TRUE and is used for configuration of operation source audit reporting. An Audit Reporter object with this flag set shall not be deletable. If creation of Audit Reporter objects is supported by the device, any created Audit Reporter objects shall have the Audit\_Source\_Reporter flag set to FALSE.

All write operations to an Audit Reporter object shall result in an audit notification by the containing device, if Audit\_Level is not NONE.

Audit Reporter objects may optionally support delaying the sending of audit notifications so that multiple notifications can be sent in a single request. The Maximum\_Send\_Delay property limits how long audit notifications may be delayed.

### 19.Y.3 Audit Notification Configuration

Any device which supports auditing shall contain at least one Audit Reporter object.

Auditing can be configured on a per Audit Reporter level and on a per object level. Group enabling and disabling of audit reporting is controlled via the Audit\_Level and Auditable\_Operations properties of the Audit Reporter objects.

Audit\_Level controls the level of audit reporting at the Audit Reporter or at the object level. When the Audit\_Level property in the Audit Reporter object is NONE, the Audit Reporter object shall not generate any audit notifications, with the exception of generating a notification when its Audit\_Level property is modified. When the Audit\_Level property in an object (other than an Audit Reporter object) is NONE, the device shall not generate any audit notifications for the object, with the exception of generating a notification when the object's Audit\_Level is modified.

The Auditable\_Operations property in an object controls which operations on the object an Audit Reporter object generates audit notifications for. If the property is not present, then the Auditable\_Operations property in the associated Audit Reporter object is used instead. Irrespective of the value of an Auditable\_Operations property, changes to an Auditable\_Operations property will always result in an audit notification by the server device if Audit\_Level is not NONE.

When a device is acting as an operation source, the Auditable\_Operations property of the Audit Reporter object, with the Audit\_Source\_Reporter property set to TRUE, controls which operations initiated by the device will be reported. The Auditable\_Operations property of the target object has no impact on the audit notifications generated by the operation source.

The Audit\_Priority\_Filter property specifies the priorities for which audit notifications will be generated for write operations. Any auditable write operation to the commandable property of an object with a priority associated with a bit set to 0 in this property shall not result in an audit notification. For example, if bit 7 of this property is set and bit 10 is not set, a manual override at priority 8 would result in an audit notification, whereas a WriteProperty request at priority 11 would not (the difference in the bit numbers versus the priorities in this example is due to bits being 0-based and priorities being 1-based). This property has no effect on the auditing of non-write operations nor on the auditing of operations on properties other than commandable properties. If this property is not present in an object with a commandable property, then no command priority based audit filtering will be applied. The purpose of this property is to allow normal control actions to occur without audit notifications being generated while ensuring that override actions are audited.

The Audit\_Level property determines which properties in the object are auditable. Irrespective of the value of this property, changes to this property shall always result in an audit notification by the server device.

**Table 19-Y1. Audit Level**

| Operation    | Comment                                                                                                                                                                                                                            |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NONE         | No operations on the object will result in audit notifications except for the noted modifications to the audit configuration.                                                                                                      |
| DEFAULT      | The value for the audit level for this object is taken from the Audit_Level property of the associated Audit Reporter object.                                                                                                      |
| AUDIT_CONFIG | Operations on configuration properties will result in audit notifications. Operations on the Present_Value will not result in audit notifications unless Present_Value has been designated as a configurable item for the product. |
| AUDIT_ALL    | Operations on all properties will result in audit notifications.                                                                                                                                                                   |

The BACnetAuditLevel enumeration is extensible providing a method for vendors to specify new audit levels for objects that require them. No extensible audit level value shall be equivalent to NONE (result in no auditable operations on the object.)

For the audit level AUDIT\_CONFIG, it is a local matter whether or not any given property in an object is considered a configuration property. This is a local matter specifically to allow implementations to make this designation based on the needs of the specific application of the product.

#### 19.Y.4 Audit Notifications

The parameters sent in an audit notification are described in Table 19-Y2.

**Table 19-Y2.** Audit Notification Parameters

| Parameter        | Data Type              | Value                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Source Timestamp | BACnetTimeStamp        | The time the operation was requested by the operation source.<br><br>This field shall not be included in operation target notifications unless the source and target are the same device.                                                                                                                                                                                                                                                       |
| Target Timestamp | BACnetTimeStamp        | The time the operation was performed by the operation target.<br><br>This field shall not be included in operation source notifications unless the source and target are the same device.                                                                                                                                                                                                                                                       |
| Source Device    | BACnetRecipient        | The device requesting the operation.<br><br>If known, the BACnetObjectIdentifier form shall be provided.                                                                                                                                                                                                                                                                                                                                        |
| Source Object    | BACnetObjectIdentifier | The object requesting the operation. This field shall not be included in operation target notifications, unless the source and target are the same device, or when there is no source object to report.                                                                                                                                                                                                                                         |
| Operation        | BACnetAuditOperation   | The operation being requested.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Source Comment   | CharacterString        | A human readable description of the operation. This value may be system generated. For systems where human comments are required for documenting actions such as required by some pharmaceutical installations, the human entered comment shall be provided in this parameter in addition to any system generated comment. For GENERAL operation audit notifications, this field shall include a system generated description of the operation. |
| Target Comment   | CharacterString        | A human readable description of the operation. For GENERAL operation audit notifications, this field shall include a system generated description of the operation.                                                                                                                                                                                                                                                                             |
| Invoke Id        | Unsigned8              | The invoke id from the service request that resulted in this audit record. This provides the ability to more easily match operation source and operation target audit records.<br><br>This field shall not be included in operations consisting of unconfirmed service requests.<br>This field shall not be included if the source and target are the same device.                                                                              |

| Parameter        | Data Type               | Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Source User Id   | Unsigned16              | <p>The user performing the operation. The value in this field depends on the user authentication model in use. See Clause 24.14 on user authentication for information on this value.</p> <p>This field shall not be included if no user identification is available.</p>                                                                                                                                                                                                                                                                                                                            |
| Source User Role | Unsigned8               | <p>The role of the user performing the operation. The value in this field depends on the user authentication model in use. See Clause 24.14 on user authentication for information on this value.</p> <p>This field shall not be included if no user role information is available.</p>                                                                                                                                                                                                                                                                                                              |
| Target Device    | BACnetRecipient         | <p>The target device of the operation.</p> <p>If known, the BACnetObjectIdentifier form shall be provided.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Target Object    | BACnetObjectIdentifier  | The target object of the operation. This field shall be absent if the target is not an object.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Target Property  | BACnetPropertyReference | The target of the operation if it is targeted at a property. This field shall be absent if the target is not a property.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Target Priority  | Unsigned(1..16)         | For operations which include a priority (such as when a command occurs) this field is present and contains the provided priority. This field may be absent if the provided priority is 16.                                                                                                                                                                                                                                                                                                                                                                                                           |
| Target Value     | ABSTRACT-SYNTAX.&Type   | <p>For operations which include a value (such as when a write occurs) this field is present and contains the new value.</p> <p>When the value being written is larger than 32 encoded octets, it is a local matter whether the target value is included in the audit notification. This allowance is included so that large audit notifications can be avoided where the data value is large.</p> <p>Audit loggers may discard this parameter value if it exceeds 500 octets when not acting as a forwarder.</p>                                                                                     |
| Current Value    | ABSTRACT-SYNTAX.&Type   | <p>When an operation includes a value (such as when a write occurs) and the current value (before the operation) of the target property is known, this field is present and contains the current property value.</p> <p>When the current value is larger than 32 encoded octets, it is a local matter whether the current value is included in the audit notification. This allowance is included so that large audit notifications can be avoided where the data value is large.</p> <p>Audit loggers may discard this parameter value if it exceeds 500 octets when not acting as a forwarder.</p> |
| Result           | Error                   | When the operation fails, this field shall be present and indicates the error that occurred.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

### 19.Y.5 Audit Operations

The 'Operation' field of an audit notification indicates the operation requested or performed. The mapping of standard BACnet services to the BACnetAuditOperation enumeration is shown in Table 19-Y3. When a service allows multiple object or property targets (e.g., ReadPropertyMultiple, WritePropertyMultiple, etc.), each target shall be reported in a separate notification.

**Table 19-Y3. Service to Audit Operation Mapping**

| Audit Operation     | BACnet Service                                                                                                 | Comment                                                                                                                                                                                                                                                                                                   |
|---------------------|----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| READ                | ReadProperty<br>ReadPropertyMultiple<br>ReadRange<br>AtomicReadFile<br>AuditLogQuery                           | When reporting a file operation, no 'Target Property' value is provided.<br><br>This setting should be used with caution as it can result in a large increase in network traffic.                                                                                                                         |
| WRITE               | WriteProperty<br>WritePropertyMultiple<br>AtomicWriteFile<br>WriteGroup<br>AddListElement<br>RemoveListElement | When reporting a file operation, no 'Target Property', 'Target Value', or 'Current Value' values are provided.<br><br>These entries can be discarded if the write operation does not result in a change (the 'Target Value' equals the 'Current Value', and re-writing the property has no side-effects). |
| CREATE              | CreateObject                                                                                                   | Initial values are not provided in a CREATE operation audit record. Initial values may be optionally reported by reporting WRITE operations.                                                                                                                                                              |
| DELETE              | DeleteObject                                                                                                   | No 'Target Property', 'Target Value', or 'Current Value' values are provided.                                                                                                                                                                                                                             |
| LIFE_SAFETY         | LifeSafetyOperation                                                                                            | The 'Target Value' shall include the 'Requesting Process Identifier' and 'Request' parameters in a BACnetLifeSafetyOperationInfo sequence.<br><br>The 'Source Comment' shall be set to, or include, the 'Requesting Source' parameter value.                                                              |
| ACKNOWLEDGE_ALARM   | AcknowledgeAlarm                                                                                               | The 'Target Value' shall include the 'Event State Acknowledged', and 'Timestamp' parameters in a BACnetAcknowledgeAlarmInfo sequence.<br><br>The 'Source Comment' shall be set to, or include, the 'Acknowledgement Source' parameter value.                                                              |
| DEVICE_DISABLE_COMM | DeviceCommunicationControl                                                                                     | Used when a DISABLE or DISABLE_INITIATION DeviceCommunicationControl operation is performed.<br><br>No 'Target Object', 'Target Property', 'Target Value', or 'Current Value' values are provided.                                                                                                        |
| DEVICE_ENABLE_COMM  | DeviceCommunicationControl                                                                                     | Used when an ENABLE DeviceCommunicationControl operation is performed, or when a DISABLE or DISABLE_INITIATION DeviceCommunicationControl operation's lifetime expires.<br><br>No 'Target Object', 'Target Property', 'Target Value', or 'Current Value' values are provided.                             |
| DEVICE_RESET        | ReinitializeDevice                                                                                             | WARMSTART and COLDSTART options of the ReinitializeDevice service. An audit record is generated when the device is reset for any reason.<br><br>No 'Target Object', 'Target Property', 'Target Value', or 'Current Value' values are provided.                                                            |

| Audit Operation  | BACnet Service                         | Comment                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DEVICE_BACKUP    | ReinitializeDevice                     | <p>START_BACKUP, ABORT_BACKUP, or END_BACKUP options of the ReinitializeDevice service.</p> <p>No 'Target Object', 'Target Property', 'Target Value', or 'Current Value' values are provided.</p>                                                                                                                                                                                                                                                                                                                                            |
| DEVICE_RESTORE   | ReinitializeDevice                     | <p>START_RESTORE, ABORT_RESTORE, END_RESTORE options of the ReinitializeDevice service.</p> <p>No 'Target Object', 'Target Property', 'Target Value', or 'Current Value' values are provided.</p>                                                                                                                                                                                                                                                                                                                                            |
| SUBSCRIPTION     | SubscribeCOV<br>SubscribeCOVProperty   | <p>The 'Target Value' parameter shall be of type BOOLEAN and indicates whether (TRUE) the operation is a subscription, or (FALSE) the operation is an unsubscription. Subscription timeouts shall be reported as unsubscriptions.</p> <p>No 'Current Value' parameter is provided.</p>                                                                                                                                                                                                                                                       |
| NOTIFICATION     | Event Notification<br>COV Notification | <p>Receipt of event notification and COV notification messages are treated as source auditable operations; whereas the generation of event notifications and COV notifications are treated as target auditable operations.</p> <p>This setting should be used with caution as it can result in a large increase in network traffic.</p> <p>The 'Target Property' shall be the Event_State property, and the 'Target Value' shall be the new Event State.</p>                                                                                 |
| AUDITING_FAILURE |                                        | <p>Used to report that audit records were dropped.</p> <p>The 'Target Timestamp' field is used to report the time of earliest record that was dropped.</p> <p>The 'Source Device' and 'Target Device' fields are set to reference the local device.</p> <p>The 'Current Value' field is set to an Unsigned count of the number of records dropped.</p>                                                                                                                                                                                       |
| NETWORK_CHANGES  |                                        | <p>Used to report changes to Network Port objects which are a result of something other than application layer BACnet services (such as BVLL messages or registration timeouts).</p>                                                                                                                                                                                                                                                                                                                                                         |
| GENERAL          |                                        | <p>Any other actions that the device needs to report to the audit log.</p> <p>Given the non-standard nature of this audit record type, it is important that information be formatted into the 'Comment' parameter.</p> <p>Where a device reports a wide variety of other actions, or there is a large variation in the expected frequency of different types of actions, it is recommended that implementers provide a method of configuring which actions result in notifications when the GENERAL operation notifications are enabled.</p> |

### 19.Y.6 High Volume Situations

Under some high traffic conditions, a device might be unable to generate audit notifications for all auditable actions due to local resource constraints. When such a condition exists, the device shall selectively drop audit records attempting to keep the most important records for distribution to the audit logger. The device shall report to the audit logger the number of dropped records using an audit notification with an operation of AUDITING\_FAILURE.

Except for notifications mandated by this standard for changes in audit settings, the relative priority of audit notifications is a local matter, it is suggested that when dropping notifications, non-modification audit notifications should be dropped first (e.g., audit operation is READ, NOTIFICATION, etc.). Audit notifications that are mandated by this standard for changes in audit settings shall be deemed to be the highest priority records and the newest of these shall be the last records to be dropped.

### 19.Y.7 Audit Logs

#### 19.Y.7.1 Audit Logger

The logging device is the central repository for audit logs. Since it is expected that audit logs will be very large, such a device is expected to have a large amount of storage dedicated to audit logging.

The general structure is that one or more Audit Log objects exist in the audit logger into which received audit notifications are placed. It is a local matter how many and which Audit Log objects any particular audit notification is placed into with the exception that, for any specific object, there is at least one Audit Log object which contains the complete audit history for the object.

Audit loggers differ from other loggers in that archiving of audit log records is accomplished by operation sources and operation targets pushing notifications to the logger and BUFFER\_READY is not used to inform archival loggers of new notification records. Also, given that auditing should not be interrupted, there is no support for the Stop\_When\_Full function.

#### 19.Y.7.2 Audit Forwarder

An audit forwarder is a specialized audit logger that is used to reduce network traffic caused by audit logging.

The audit forwarder collects audit notifications from multiple requests into an Audit Log object and then forwards them to a parent audit logger. Once forwarded successfully, audit notifications may be removed from the forwarder's Audit Log object.

Audit forwarders are not required to support the audit querying functionality and as such are not considered a source of audit information for audit reporting. The audit forwarder is simply used for reducing the impact of audit reporting on network usage.

Audit forwarders shall not discard, or change, parameter values in audit notifications from other devices when forwarding the notifications.

While audit forwarder functionality can be placed in any device in the system, it is recommended that it be placed in routers as this minimizes the impact on network traffic.

#### 19.Y.7.3 Audit Logger Hierarchies

When the operation source and operation target devices involved in an auditable operation report audit notifications to different audit loggers, there will be two distinct records of the operation in two different audit loggers. If there is to be a consolidated log, then a hierarchy of audit loggers shall be setup. A complete log with all audit record pairs will only be available in the topmost audit logger in the hierarchy. The hierarchy is achieved through the Member\_Of property of the Audit Log object.

### 19.Y.8 Security of Audit Notifications

The BACnet audit logging framework does not provide any security. Security of the network shall be achieved through other means such as the security framework described in Clause 24.

Given that values are stored in Audit Logs, care should be taken in configuring access to Audit Log objects so that information that needs to be restricted can be.

[Insert into **Clause 3.2**]

**persistent:** when used to describe property values, the property value is stored in non-volatile storage so as to survive device resets and power failures.

[Modify **Table 12-13**]

|                                     |                                                                                                    |                        |  |
|-------------------------------------|----------------------------------------------------------------------------------------------------|------------------------|--|
| <i>Audit_Notification_Recipient</i> | <i>BACnetRecipient</i>                                                                             | <i>O</i> <sup>22</sup> |  |
| ...                                 |                                                                                                    |                        |  |
| ...                                 |                                                                                                    |                        |  |
| <sup>22</sup>                       | If present, this property shall be writable.                                                       |                        |  |
| ...                                 |                                                                                                    |                        |  |
| <sup>x</sup>                        | <i>This property shall be present if, and only if, the device supports <u>audit reporting</u>.</i> |                        |  |

[Add new **Clause 12.11.X1**]

#### **12.11.X1    Audit\_Notification\_Recipient**

This property, of type BACnetRecipient, specifies the audit logging device to which audit notifications are sent.

When this property is changed, an audit notification shall be generated and it shall be either globally broadcast, or sent to both the original audit notification recipient and the new audit notification recipient.

If this property is present, it shall be writable.

[Add new **Clause 12.X**]

#### **12.X Audit Reporter Object Type**

The Audit Reporter object type defines a standardized object whose properties represent the externally visible audit settings for a BACnet device which reports auditable actions. BACnet devices which report auditable actions shall contain at least one Audit Reporter object. A detailed description of BACnet audit reporting can be found in Clause 19.Y.

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

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

**Table 12-X.** Properties of the Audit Reporter Object Type

| Property Identifier            | Property Datatype                          | Conformance Code |
|--------------------------------|--------------------------------------------|------------------|
| Object_Identifier              | BACnetObjectIdentifier                     | R                |
| Object_Name                    | CharacterString                            | R                |
| Object_Type                    | BACnetObjectType                           | R                |
| Description                    | CharacterString                            | O                |
| Status_Flags                   | BACnetStatusFlags                          | R                |
| Reliability                    | BACnetReliability                          | R                |
| Event_State                    | BACnetEventState                           | R                |
| Audit_Level                    | BACnetAuditLevel                           | R                |
| Audit_Source_Reporter          | BOOLEAN                                    | R                |
| Auditable_Operations           | BACnetAuditOperationFlags                  | R                |
| Audit_Priority_Filter          | BACnetPriorityFilter                       | R                |
| Issue_Confirmed_Notifications  | BOOLEAN                                    | R                |
| Monitored_Objects              | BACnetARRAY[N] of BACnetObjectSelector     | O                |
| Maximum_Send_Delay             | Unsigned                                   | O <sup>1,2</sup> |
| Send_Now                       | BOOLEAN                                    | O <sup>1,2</sup> |
| Event_Detection_Enable         | BOOLEAN                                    | O <sup>3,5</sup> |
| Notification_Class             | Unsigned                                   | O <sup>3,5</sup> |
| Event_Enable                   | BACnetEventTransitionBits                  | O <sup>3,5</sup> |
| Acked_Transitions              | BACnetEventTransitionBits                  | O <sup>3,5</sup> |
| Notify_Type                    | BACnetNotifyType                           | O <sup>3,5</sup> |
| Event_Time_Stamps              | BACnetARRAY[3] of BACnetTimeStamp          | O <sup>3,5</sup> |
| Event_Message_Texts            | BACnetARRAY[3] of CharacterString          | O <sup>4,5</sup> |
| Event_Message_Texts_Config     | BACnetARRAY[3] of CharacterString          | O <sup>5</sup>   |
| Reliability_Evaluation_Inhibit | BOOLEAN                                    | O                |
| Property_List                  | BACnetARRAY[N] of BACnetPropertyIdentifier | R                |
| Tags                           | BACnetARRAY[N] of BACnetNameValue          | O                |
| Profile_Location               | CharacterString                            | O                |
| Profile_Name                   | CharacterString                            | O                |

<sup>1</sup> If present, these properties shall be writable.

<sup>2</sup> If the object supports delaying of audit notifications, then these properties shall be present.

<sup>3</sup> These properties are required if the object supports intrinsic reporting.

<sup>4</sup> This property, if present, is required to be read-only.

<sup>5</sup> These properties shall be present only if the object supports intrinsic 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 AUDIT\_REPORTER.

### 12.X.4 Description

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

### 12.X.5 Status\_Flags

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

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

where:

|                |                                                                                                                                      |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------|
| IN_ALARM       | Logical 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     | Logical FALSE (0).                                                                                                                   |
| OUT_OF_SERVICE | Logical FALSE (0).                                                                                                                   |

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

### 12.X.6 Reliability

The Reliability property, of type BACnetReliability, provides an indication that the properties of the Audit Reporter object are in a consistent state and whether the object is operating properly.

If this Audit Reporter object is enabled and configured to monitor one or more objects that another enabled Audit Reporter object is also configured to monitor, then this property shall have the value CONFIGURATION\_ERROR.

If the Audit Reporter object is enabled and the device is unable to resolve the Audit\_Notification\_Recipient device, then this property shall have the value CONFIGURATION\_ERROR.

If the Audit Reporter object fails to successfully send audit notifications, or fails to receive acknowledgements when using ConfirmedAuditNotification requests, the Reliability property shall be set to COMMUNICATION\_FAILURE. The existence of this reliability condition shall not stop the object from attempting to send audit notifications. It is a local matter whether or not the object will retry sending of audit notifications.

### 12.X.7 Event\_State

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

### 12.X.8 Audit\_Level

This property, of type BACnetAuditLevel, specifies the level of auditing to perform for all objects this Audit Reporter object reports for which do not have an Audit\_Level property, or for which the Audit\_Level property is set to DEFAULT.

This property shall not have the value DEFAULT.

If not configurable, this property shall not have the value NONE.

For details on auditing and the use of this property, see Clause 19.Y.2.

#### 12.X.9 Audit\_Source\_Reporter

This read-only property, of type BOOLEAN, indicates whether (TRUE) or not (FALSE) this audit reporter object is the reporter object which controls operation source audit reporting.

If this property is TRUE, this object generates notifications for auditable operations initiated by this device.

For details on auditing and the use of this property, see Clause 19.Y.2.

#### 12.X.10 Auditable\_Operations

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

If 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.

For details on auditing and the use of this property, see Clause 19.Y.3.

#### 12.X.11 Audit\_Priority\_Filter

This property, of type BACnetPriorityFilter, specifies the auditable command priorities for write operations on the commandable property of the objects for which this object performs audit reporting. Bits which are set indicate the priorities for which audit notifications will be generated. The bit number is 1 less than the priority it is associated with (bit 7 is associated with priority 8).

For details on auditing and the use of this property, see Clause 19.Y.3.

#### 12.X.12 Issue\_Confirmed\_Notifications

This property, of type BOOLEAN, shall convey whether confirmed (TRUE) or unconfirmed (FALSE) audit notifications shall be issued.

#### 12.X.13 Monitored\_Objects

This property, of type BACnetARRAY of BACnetObjectSelector, is used to indicate the objects that the Audit Reporter object will generate notifications for.

If the array is of length 0, or all entries in the array are NULL entries, then the Audit Reporter object does not generate notifications for any objects in the device.

Entries of type BACnetObjectIdentifier indicate specific objects that are reportable by the Audit Reporter object. Entries of type BACnetObjectType indicate that all objects of the specified type are reportable by the Audit Reporter object. Entries of type NULL are ignored.

If there are multiple Audit Reporter objects which are configured to report for a specific object, only the one with the lowest object instance shall generate notifications for the object. This allows a generic catch-all Audit Reporter object to be created with a high instance number and lower instance number Audit Reporter objects which provide customized audit reporting for specific objects.

#### 12.X.14 Maximum\_Send\_Delay

This property, of type Unsigned, specifies the maximum amount of time, in seconds, that the Audit Reporter object will delay sending audit notifications for the purpose of batch sending notifications.

The value in this property does not mandate that notifications shall be delayed, but rather allows the object to delay sending of notifications by up to the specified number of seconds.

This property shall support all values in the range 0 to 3600.

**12.X.15 Send\_Now**

This property, of type BOOLEAN, when written to TRUE, forces the Audit Reporter object to send any and all audit notifications that are being delayed. After all delayed audit notifications are sent, the Audit Reporter object resets the property to FALSE.

It is a local matter whether or not the Audit Reporter object stops attempting to immediately send its delayed notifications if the value is written to FALSE.

**12.X.16 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.17 Notification\_Class**

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

**12.X.18 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.19 Acked\_Transitions**

This read-only property, of type BACnetEventTransitionBits, shall convey three flags that separately indicate the acknowledgment state for TO\_OFFNORMAL, TO\_FAULT, and TO\_NORMAL events (see Clause 13.2.2.1.5). Each flag shall have the value TRUE if no event of that type has ever occurred for the object.

**12.X.20 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.21 Event\_Time\_Stamps**

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

**12.X.22 Event\_Message\_Texts**

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

**12.X.23 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.24 Reliability\_Evaluation\_Inhibit

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

When reliability-evaluation is disabled, the Reliability property shall have the value NO\_FAULT\_DETECTED unless Out\_Of\_Service is TRUE and an alternate value has been written to the Reliability property.

#### 12.X.25 Property\_List

This read-only property is a BACnetARRAY of property identifiers, one property identifier for each property that exists within the object. The Object\_Name, Object\_Type, Object\_Identifier, and Property\_List properties are not included in the list.

#### 12.X.26 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 tag 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.27 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.28 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 new Clause 12.Y]

### 12.Y Audit Log Object Type

An Audit Log object combines audit notifications from operation sources and operation targets and stores the combined record in an internal buffer for subsequent retrieval. Each timestamped buffer entry is called an audit log "record."

Each Audit Log object maintains an internal, persistent, optionally fixed-size log buffer. This log buffer fills or grows as audit log records are added. If the log buffer becomes full, the least recent log records are overwritten when new log records are added. Log buffers are transferred as a list of BACnetAuditLogRecord values using the ReadRange and AuditLogQuery services. Each log record in the log buffer has an implied sequence number that is equal to the value of the Total\_Record\_Count property immediately after the record is added. See Clause 19.Y for a full description of how audit notifications are added to audit logs.

As records are added into the log, the Audit Log object will scan existing entries for a matching record. A record is a match if:

- 1) the record contains the timestamp for the opposite actor (the record contains the operation source timestamp when merging in an operation target notification and vice-versa);
- 2) the operation-source, operation, invoke-id, target-device, target-property, are all equal;
- 3) if the user-id, user-role, target-value fields are provided in both notifications then they are equal; and
- 4) if the source-timestamp and target-timestamp values are approximately equal ( $\pm \text{APDU\_Timeout} * 2$ ).

If a match is found, the existing log record is updated. Otherwise, a new record is created. If a match is found, and it already contains both an operation source and an operation target portion, then the notification is dropped. When creating a new record, those fields which are not supplied in the notification (such as the 'Source Timestamp' when a server notification is received) shall be absent from the record. When updating an existing record, those fields not supplied in the original notification are updated from the new notification, if present. For the 'Current Value' field, a value provided by the operation target device shall always take precedence over a value provided by an operation source device. As such, if the values provided in the peer notifications differ, the operation target value shall be the one used in the record.

Logging may be enabled and disabled through the Enable property. Audit Log enabling and disabling is recorded in the audit log buffer.

Unlike other log objects, Audit Log objects do not use the BUFFER\_READY event algorithm.

The acquisition of log records by remote devices has no effect upon the state of the Audit Log object itself. This allows completely independent, but properly sequential, access to its log records by all remote devices. Any remote device can independently update its log records at any time.

Audit Log objects may optionally support forwarding of audit notifications to “parent” audit logs. This functionality improves the reliability of the audit system by allowing intermediaries to buffer audit notifications in the case where the ultimate audit logger is offline for a short period of time. It is expected that intermediaries be capable of storing a larger number of records than devices which report auditable actions. It is also useful for buffering of audit notifications so they can be sent in bulk to the parent audit log. When operating in this mode, with the Delete\_On\_Forward property set to TRUE, the object is not required to perform audit notification matching and combining.

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

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

**Table 12-Y.** Properties of the Audit Log Object Type

| Property Identifier            | Property Datatype                          | Conformance Code |
|--------------------------------|--------------------------------------------|------------------|
| Object_Identifier              | BACnetObjectIdentifier                     | R                |
| Object_Name                    | CharacterString                            | R                |
| Object_Type                    | BACnetObjectType                           | R                |
| Property_List                  | BACnetARRAY[N] of BACnetPropertyIdentifier | R                |
| Description                    | CharacterString                            | O                |
| Status_Flags                   | BACnetStatusFlags                          | R                |
| Event_State                    | BACnetEventState                           | R                |
| Reliability                    | BACnetReliability                          | O                |
| Enable                         | BOOLEAN                                    | W                |
| Buffer_Size                    | Unsigned32                                 | R                |
| Log_Buffer                     | BACnetLIST of BACnetAuditLogRecord         | R                |
| Record_Count                   | Unsigned64                                 | R                |
| Total_Record_Count             | Unsigned64                                 | R                |
| Member_Of                      | BACnetDeviceObjectReference                | O <sup>1</sup>   |
| Delete_On_Forward              | BOOLEAN                                    | O <sup>1,2</sup> |
| Issue_Confirmed_Notifications  | BOOLEAN                                    | O <sup>1</sup>   |
| Event_Detection_Enable         | BOOLEAN                                    | O <sup>3,4</sup> |
| Notification_Class             | Unsigned                                   | O <sup>3,4</sup> |
| Event_Enable                   | BACnetEventTransitionBits                  | O <sup>3,4</sup> |
| Acked_Transitions              | BACnetEventTransitionBits                  | O <sup>3,4</sup> |
| Notify_Type                    | BACnetNotifyType                           | O <sup>3,4</sup> |
| Event_Time_Stamps              | BACnetARRAY[3] of BACnetTimeStamp          | O <sup>3,4</sup> |
| Event_Message_Texts            | BACnetARRAY[3] of CharacterString          | O <sup>4,5</sup> |
| Event_Message_Texts_Config     | BACnetARRAY[3] of CharacterString          | O <sup>4</sup>   |
| Event_Detection_Enable         | BOOLEAN                                    | O <sup>3,4</sup> |
| Reliability_Evaluation_Inhibit | BOOLEAN                                    | O <sup>6</sup>   |
| Audit_Level                    | BACnetAuditLevel                           | O <sup>7</sup>   |
| Auditable_Operations           | BACnetAuditOperationFlags                  | O <sup>7</sup>   |
| Tags                           | BACnetARRAY[N] of BACnetNameValue          | O                |
| Profile_Location               | CharacterString                            | O                |
| Profile_Name                   | CharacterString                            | O                |

<sup>1</sup> These properties shall be present if the object supports forwarding.

<sup>2</sup> This property shall be read-only and TRUE if the object does not support matching and combining of records.

<sup>3</sup> These properties are required if the object supports intrinsic reporting.

<sup>4</sup> These properties shall be present only if the object supports intrinsic reporting.

<sup>5</sup> This property, if present, is required to be read-only.

<sup>6</sup> If this property is present, then the Reliability property shall be present.

<sup>7</sup> This property shall be present only if the device supports audit reporting.

### 12.Y.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.Y.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.Y.3 Object\_Type

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

### 12.Y.4 Property\_List

This read-only property is a BACnetARRAY of property identifiers, one property identifier for each property that exists within the object. The Object\_Name, Object\_Type, Object\_Identifier, and Property\_List properties are not included in the list.

### 12.Y.5 Description

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

### 12.Y.6 Status\_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of an Audit Log object. The IN\_ALARM and FAULT flags are associated with the values of other properties of this object. A more detailed status may be determined by reading the properties that are linked to these flags. The relationship between individual flags is not defined by the protocol. The four flags are

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

where:

|                |                                                                                                                                      |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------|
| IN_ALARM       | Logical FALSE (0) if the Event_State property has a value of NORMAL, otherwise logical TRUE (1).                                     |
| FAULT          | Logical TRUE (1) if the Reliability property 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).                                                                                   |

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

### 12.Y.7 Event\_State

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

### 12.Y.8 Reliability

The Reliability property, of type BACnetReliability, provides an indication that the properties of the Audit Log object are in a consistent state and whether the object is operating properly.

If the Audit Log object is enabled and configured to forward audit notifications, and the device is unable to resolve the Member\_Of device, then this property shall have the value CONFIGURATION\_ERROR.

If the Audit Log object, when configured to forward audit notifications, fails to successfully send audit notifications, or fails to receive acknowledgements when using ConfirmedAuditNotification requests, the Reliability property shall be set to COMMUNICATION\_FAILURE. The existence of this reliability condition shall not stop the object from attempting to send audit notifications. It is a local matter whether or not the object will retry sending of audit notifications.

### 12.Y.9 Enable

This property, of type BOOLEAN, indicates and controls whether (TRUE) or not (FALSE) logging is enabled. Logging occurs if and only if Enable is TRUE. Log\_Buffer records of type log-status are recorded without regard to the value of the Enable property.

### 12.Y.10 Buffer\_Size

This property, of type Unsigned32, shall specify the maximum number of log records the log buffer may hold. If writable, it may not be written when Enable is TRUE. The disposition of existing log records when Buffer\_Size is written is a local matter.

For products that support very large log objects, the value  $2^{32}-1$  is reserved to indicate that the buffer size is unknown and is constrained solely by currently available resources.

### 12.Y.11 Log\_Buffer

This property, of type BACnetLIST of BACnetAuditLogRecord, is a list of timestamped log records of datatype BACnetAuditLogRecord, each of which conveys the audit notification parameters or status changes in the Audit Log object. Each log record has data fields as follows:

**Timestamp** The local date and time that the entry was placed into the audit log.

**Log Datum** The audit notification information, or a change in status or operation of the Audit Log object itself.

The choices available for the 'Log Datum' are listed below:

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| log-status         | This choice represents a change in the status or operation of the Audit Log object. Whenever one of the events represented by the flags listed below occurs, a log record shall be appended to the buffer.                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|                    | LOG_DISABLED                                                                                                                                                                                                                                                                                                                                                                                                                                 | This flag is changed whenever collection of log records by the Audit Log object is enabled or disabled. It shall be TRUE if Enable is FALSE; otherwise it shall be FALSE.                                                                                                                                                                                                                                                                                                         |
|                    | BUFFER_PURGED                                                                                                                                                                                                                                                                                                                                                                                                                                | This flag shall be set to TRUE whenever the log buffer is cleared. After this value is recorded in the log buffer, the subsequent immediate change to FALSE shall not be recorded. A log record indicating the purging of the log buffer shall be placed into the log buffer even if logging is disabled, the local time is outside of the time range defined by the Start_Time and Stop_Time properties, or the log buffer was completely empty before the request for clearing. |
|                    | LOG_INTERRUPTED                                                                                                                                                                                                                                                                                                                                                                                                                              | This flag indicates that the collection of log records by the Audit Log object was interrupted by a power failure, device reset, object reconfiguration, or other such disruption, such that samples prior to this log record might have been missed.                                                                                                                                                                                                                             |
| audit-notification | This choice represents an audit notification that was received. It consists of the audit information from the operation source and operation target generated ConfirmedAuditNotification and/or UnconfirmedAuditNotification requests. If the audit notification was generated locally, this choice shall hold what would be received if the Audit Log object existed on a remote device.                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| time-change        | This choice represents a change in the clock setting in the device, records the number of seconds and fraction of a second by which the clock changed. If, and only if, the number is not known, such as when the clock is initialized for the first time, the value recorded shall be 0.0. This log record shall be recorded after changing the local time of the device, and the timestamp shall reflect the new local time of the device. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |