

BSR/ASHRAE Addendum p
to ANSI/ASHRAE Standard 135-2008

Public Review Draft

ASHRAE® Standard

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

First Public Review (**March 2009**)
(Draft Shows Changes to the Current
Standard)

This draft has been recommended for public review by the responsible project committee. To submit a comment on this proposed addendum, go to the ASHRAE website at <http://www.ashrae.org/technology/page/331> and access the online comment database. The draft is subject to modification until it is approved for publication by the Board of Directors and ANSI. Until this time, the current edition of the standard (as modified by any published addenda on the ASHRAE web site) remains in effect. The current edition of any standard may be purchased from the ASHRAE Bookstore @ <http://www.ashrae.org> or by calling 404-636-8400 or 1-800-527-4723 (for orders in the U.S. or Canada).

This standard is under continuous maintenance. To propose a change to the current standard, use the change submittal form available on the ASHRAE web site @ <http://www.ashrae.org>.

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

© **March 20, 2009**. This draft is covered under ASHRAE copyright. Permission to reproduce or redistribute all or any part of this document must be obtained from the ASHRAE Manager of Standards, 1791 Tullie Circle, NE, Atlanta, GA 30329-2305. Phone: 404-636-8400, Ext1125. Fax: 404-321-5478. E-mail: standards.section@ashrae.org.

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



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

FOREWORD

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

135-2008p-1. Add a new Global Group object type, p. 1.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2008 and Addenda is indicated through the use of *italics*, while deletions are indicated by ~~strike through~~. Where entirely new subclauses are proposed to be added, plain type is used throughout. Only this new and deleted text is open to comment at this time. All other material in this addendum is provided for context only and is not open to public review comment except as it relates to the proposed changes.

135-2008p-1. Add a new Global Group object type.

Rationale
 There is need for a standard object type similar to the Group object type except that it can provide a collection of information from objects in a number of BACnet devices and it can also deliver that information in an intrinsic event notification when any of the group member objects enters a non-NORMAL state.

Addendum 135-2008p-1

[Add new Clause 12.Y, p. 288]

12.Y Global Group Object Type

The Global Group object type defines a standardized object whose properties represent a collection of other objects and one or more of their properties. A Global Group object is used to simplify the exchange of information between BACnet devices by providing a shorthand way to specify all members of the group at once. A Global Group object may be formed using any combination of object types except other Group and Global Group object types.

A Global Group object differs from a Group object in two ways: its members can be from anywhere in the BACnet internetwork and it supports intrinsic event reporting. If the Event_State of one of the objects that is a member of the group changes to a non-normal state, the Global Group object can initiate an event notification message conveying the values of all of the members of the group. This provides a mechanism to define an arbitrarily large set of property values that are made available when an event occurs.

The Global Group object and its properties are summarized in Table 12-Y1 and described in detail in this subclause.

Table 12-Y1. Properties of the Global Group Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Group_Members	BACnetARRAY[N] of BACnetDeviceObjectPropertyReference	R
Group_Member_Names	BACnetARRAY[N] of CharacterString	O
Present_Value	BACnetARRAY[N] of BACnetPropertyAccessResult	R
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Member_Status_Flags	BACnetStatusFlags	R
Reliability	BACnetReliability	O
Out_of_Service	BOOLEAN	R
Update_Interval	Unsigned	O
Requested_Update_Interval	Unsigned	O
Time_Delay	Unsigned	O ¹
Notification_Class	Unsigned	O ¹
Event_Enable	BACnetEventTransitionBits	O ¹
Acked_Transitions	BACnetEventTransitionBits	O ¹
Notify_Type	BACnetNotifyType	O ¹
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	O ¹
COV_Period	Unsigned	R
Profile_Name	CharacterString	O

¹ These properties are required if the object supports intrinsic 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 GLOBAL_GROUP.

12.Y.4 Description

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

12.Y.5 Group_Members

This property is a BACnetARRAY of BACnetDeviceObjectPropertyReference that defines the members of the group. If the optional device identifier is not present for a particular group member, then that object shall reside in the same device that maintains the Global Group object. If the Group_Members property is writable using WriteProperty services, then the object shall support group members that are outside the device that maintains the Global Group object.

Nesting of group objects is not permitted; that is, Group_Members shall not refer to the Present_Value property of a Group object or a Global Group object.

12.Y.5.1 Resizing Group_Member_Names Array by Writing the Group_Members Property

The size of the Group_Members array shall be maintained so that it has the same size as the Group_Member_Names array and the Present_Value array. If the Group_Members property is writable and the size of the array is reduced, the Group_Members, Group_Member_Names, and Present_Value arrays shall all be truncated to the new reduced size. If the Group_Members property is writable and the size of the array is increased, the Group_Members, Group_Member_Names, and Present_Value arrays shall all be increased to the new expanded size and the new array elements initialized according to the requirements of each property. See 12.Y.5.3, 12.Y.6.3, and 12.Y.7.1.

12.Y.5.2 Resizing Group_Members Array by Writing the Group_Member_Names Property

The size of the Group_Members array shall be maintained so that it has the same size as the Group_Member_Names array and the Present_Value array. If the size of the Group_Member_Names array is changed, there shall be a corresponding change to the size of the Group_Members array. See 12.Y.6.1.

12.Y.5.3 Initializing New Array Elements When the Array Size is Increased

If the size of the Group_Members array is increased by writing to the size of either the Group_Members or Group_Member_Names property, the new array entries shall be initialized by setting the 'Device Identifier' parameter of the BACnetDeviceObjectPropertyReference to be a Device object with an instance number of 4194303, indicating that the value is not initialized. The initial value of the other parameters is a local matter except that they must be of the correct datatype.

12.Y.6 Group_Member_Names

This optional property is a BACnetARRAY of character strings representing a descriptive name for the members of the Global Group. The number of names matches the number of members defined in Group_Members. The array index of the name shall match the array index of the corresponding group member.

12.Y.6.1 Resizing Group_Members Array by Writing the Group_Member_Names Property

The size of the Group_Member_Names array shall be maintained so that it has the same size as the Group_Members array and the Present_Value array. If the Group_Member_Names property is writable and the size of the array is reduced, the Group_Members, Group_Member_Names, and Present_Value arrays shall all be truncated to the new reduced size. If the Group_Member_Names property is writable and the size of the array is increased, the Group_Members, Group_Member_Names, and Present_Value arrays shall all be increased to the new expanded size and the new array elements initialized according to the requirements of each property. See 12.Y.5.3, 12.Y.6.3, and 12.Y.7.1.

12.Y.6.2 Resizing Group_Member_Names Array by Writing the Group_Members Property

The size of the Group_Member_Names array shall be maintained so that it has the same size as the Group_Members array and the Present_Value array. If the size of the Group_Members array is changed, there shall be a corresponding change to the size of the Group_Member_Names array. See 12.Y.5.1.

12.Y.6.3 Initializing New Array Elements When the Array Size is Increased

If the size of the Group_Member_Names array is increased by writing to the size of either the Group_Members or Group_Member_Names property, the new array entries shall be initialized with empty strings.

12.Y.7 Present_Value

This property is a read only BACnetARRAY of BACnetPropertyAccessResult that contains the values of all the properties specified in the Group_Members property. The array index of the Present_Value shall match the corresponding array index in Group_Members. This is a "read only" property; it cannot be used to write a set of values to the members of the group. The Present_Value data shall be stored locally. If the Present_Value, or a portion of the Present_Value, is acquired periodically and the Requested_Update_Interval property is present, then an attempt shall be made to update the Present_Value within this time interval. If the Present_Value, or a portion of the Present_Value, is acquired periodically and the Requested_Update_Interval is not present, then the update interval is a local matter. When updating the Present_Value, if a group member's property value cannot be acquired, a property access error shall be stored in the access result for that member of the group. If a property access error was returned when attempting to update the group member's property value, then that access error shall be the one stored in the access result. Otherwise, the choice of property access error to store shall be a local matter.

The Present_Value array shall be maintained at the same size as the Group_Members array. If the Group_Members property is writable and the size of the array is reduced, the Present_Value array shall be truncated to match. If the Group_Members property is writable and the size of the array is increased, the Present_Value array shall be increased in size to match with the value of the new array elements being determined through the same mechanism that is used to update the values. If a specific element in the Group_Members property changes, then the corresponding element in the Present_Value array shall be updated through the same mechanism that is used to update the values.

The Present_Value may be updated based on COV notifications, polling, or a combination of the two.

If the number of changed properties that need to be conveyed is too large to be encoded into a single message, then multiple UnconfirmedCOVNotifications shall be sent, grouping as many properties as will fit into each message.

If the Event_Values parameter that needs to be conveyed is too large to be encoded into a single message, then the Event_Values parameter shall be modified based on the rules specified in Table 13-3.

The value of the Present_Value property shall continue to be updated regardless of the value of the Reliability property.

12.Y.7.1 Initializing New Array Elements When the Array Size is Increased

If the size of the Present_Value array is increased by writing to the size of either the Group_Members or Group_Member_Names property, the new array entries shall be initialized with the Access Result parameter having a value of type PropertyAccessError, with an Error Class of PROPERTY and an Error Code of VALUE_NOT_INITIALIZED. The other parameters shall have values consistent with the corresponding entry in the Group_Members array.

12.Y.8 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of the Global Group 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 | Logical TRUE (1) if the global group has been overridden by some mechanism local to the BACnet Device. In this context "overridden" is taken to mean that the Event_State property is no longer tracking changes to the Event_State of group member objects and the Reliability property is no longer a reflection of the result of any internal algorithm for determining the reliability of the Global Group object. Otherwise, the value is logical FALSE (0). |
| OUT_OF_SERVICE | Logical TRUE (1) if the Out_Of_Service property has a value of TRUE, otherwise logical FALSE (0). |

12.Y.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. If the object supports intrinsic reporting, then the Event_State property shall indicate the event state of the object. If the object does not support intrinsic reporting, then the value of this property shall be NORMAL. If the Reliability property is present and does not have a value of NO_FAULT_DETECTED, then the value of this property shall be FAULT. Changes in the Event_State property to the value FAULT are considered to be "fault" events.

12.Y.10 Member_Status_Flags

The Member_Status_Flags property is a logical combination of all the Status_Flags properties contained in the Present_Value. The logical combination means that each of the flags in this property (IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE) is TRUE if and only if the corresponding flag is set in any of the Status_Flags property values in the Present_Value property. This property shall be updated whenever new Status_Flags property values are updated in the Present_Value.

12.Y.10.1 Conditions for Generating a TO-OFFNORMAL Event

A TO-OFFNORMAL event is generated under these conditions:

- (a) the IN_ALARM flag of the Member_Status_Flags property must remain equal to TRUE for a minimum period of time, specified by the Time_Delay property, and
- (b) the TO-OFFNORMAL flag must be enabled in the Event_Enable property.

When a TO-OFFNORMAL event is generated, the Global Group object shall return the most recently stored Present_Value as a parameter in the event notification message.

12.Y.10.2 Conditions for Generating a TO-NORMAL Event

A TO-NORMAL event is generated if the TO-NORMAL flag is enabled in the Event_Enable property, and either:

- (a) the IN_ALARM flag of the Member_Status_Flags property remains equal to FALSE for a minimum period of time, specified by the Time_Delay property, or
- (b) the Reliability property changes to the value NO_FAULT_DETECTED.

When a TO-NORMAL event is generated, the Global Group object shall return the most recently stored Present_Value as a parameter in the event notification message.

12.Y.10.3 Conditions for Generating a TO-FAULT Event

A TO-FAULT event is generated if the TO-FAULT flag is enabled in the Event_Enable property and the Reliability property changes to a value other than NO_FAULT_DETECTED.

When a TO-FAULT event is generated, the Global Group object shall return the most recently stored Present_Value as a parameter in the event notification message.

12.Y.11 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the Present_Value is "reliable" as far as the BACnet Device or operator can determine. If the FAULT flag of the Member_Status_Flags has a value of TRUE, then the value of this property shall be MEMBER_FAULT. If one or more group member values cannot be updated because of a communication failure, the value of this property shall be COMMUNICATION_FAILURE. If the conditions for a MEMBER_FAULT and a COMMUNICATION_FAILURE are both present, the selection of which value to use is a local matter. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, MEMBER_FAULT, COMMUNICATION_FAILURE, UNRELIABLE_OTHER}.

12.Y.12 Out_of_Service

This property, of type BOOLEAN, indicates and controls whether (TRUE) or not (FALSE) the Present_Value property is updated to track the values of the group members.

12.Y.13 Update_Interval

This optional property, of type Unsigned, provides an indication of the actual period of time-between updates to Present_Value, measured in hundredths of a second. The method used to calculate Update_Interval is a local matter.

12.Y.14 Requested_Update_Interval

This optional property, of type Unsigned, indicates the requested period of time between updates to Present_Value, measured in hundredths of a second when the object is not out-of-service.

12.Y.15 Time_Delay

This property, of type Unsigned, shall specify the minimum period of time in seconds during which the Event_State of one of the group members must have a value different from NORMAL before a TO-OFFNORMAL event is generated. It is also the minimum period of time in seconds during which the Event_State of every member of the group must remain equal to NORMAL before a TO-NORMAL event is generated. This property is required if intrinsic reporting is supported by this object.

12.Y.16 Notification_Class

This property, of type Unsigned, shall specify the notification class to be used when handling and generating event notifications for this object. The Notification_Class property implicitly refers to a Notification Class object that has a Notification_Class property with the same value. This property is required if intrinsic reporting is supported by this object.

12.Y.17 Event_Enable

This property, of type BACnetEventTransitionBits, shall convey three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. This property is required if intrinsic reporting is supported by this object.

12.Y.18 Acked_Transitions

This property, of type BACnetEventTransitionBits, shall convey three flags that separately indicate the receipt of acknowledgments for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. These flags shall be cleared upon the occurrence of the corresponding event and set under any of these conditions:

- (a) upon receipt of the corresponding acknowledgment;
- (b) upon the occurrence of the event if the corresponding flag is not set in the Event_Enable property (meaning event notifications will not be generated for this condition and thus no acknowledgment is expected);
- (c) upon the occurrence of the event if the corresponding flag is set in the Event_Enable property and the corresponding flag in the Ack_Required property of the Notification Class object implicitly referenced by the Notification_Class property of this object is not set (meaning no acknowledgment is expected).

This property is required if intrinsic reporting is supported by this object.

12.Y.19 Notify_Type

This property, of type BACnetNotifyType, shall convey whether the notifications generated by the object should be Events or Alarms. This property is required if intrinsic reporting is supported by this object.

12.Y.20 Event_Time_Stamps

This optional property, of type BACnetARRAY[3] of BACnetTimeStamp, shall convey the times of the last event notifications for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events, respectively. Time stamps of type Time or Date shall have 'FF' in each octet, and Sequence number time stamps shall have the value 0 if no event notification of that type has been generated since the object was created. This property is required if intrinsic reporting is supported by this object.

12.Y.21 COV_Period

The COV_Period property, of type Unsigned, shall indicate the amount of time in seconds between the periodic COV notifications performed by this object. COV notifications convey the value of the Present_Value and Member_Status_Flags properties. If the value of COV_Period is zero, then periodic UnconfirmedCOVNotification messages shall not be transmitted. This property is required if COV reporting is supported by this object.

12.Y.22 Profile_Name

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

A profile defines a set of additional properties, behavior, and/or requirements for this object beyond those specified here. This standard defines only the format of the names of profiles. The definition of the profiles themselves is outside the scope of this standard.

[Add entry to table in Clause 12, p. 130]

12 MODELING CONTROL DEVICES AS A COLLECTION OF OBJECTS

...	<code>NO_FAULT_DETECTED</code>	The present value is reliable; that is, no other fault (enumerated below) has been detected.
...	<code>MEMBER_FAULT</code>	... <i>Indicates that the group includes one or more Status_Flags properties whose FAULT flag value is equal to TRUE.</i>

[Add entry to table in Clause 12, p. 131]

Clause	Description	Page
...
12.Y	Global Group Object Type	p. xxx

[Change Clause 21, **BACnetReliability** production, p. 472]

```

BACnetReliability ::= ENUMERATED {
    ...
    communication-failure      (12),
    member-fault                (n),
    ...
}
    
```

[Change Clause 12.14.5, p.195]

12.14.5 List_Of_Group_Members

This property is a list of one or more read access specifications, which defines the members of the group that shall be referenced when this object is specified in a protocol transaction. Each read access specification shall consist of two parts: 1) an `Object_Identifier` and 2) a `List Of Property References`. All members of the group shall be objects that reside in the same device that maintains the `Group` object. See the ASN.1 production for `ReadAccessSpecification` in Clause 21.

Nesting of group objects is not permitted; that is, the `Group_Members` shall not refer to the `Present_Value` property of a `Group` object or a *Global Group object*.

[Add entry and footnote to **Table 13-1**, page 291]

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

Object Type	Criteria	Properties Reported
<i>Global Group</i> ¹	<i>If Present_Value changes at all or Member_Status_Flags changes at all or A periodic update of Present_Value occurs</i>	<i>Elements of Present_Value, Member_Status_Flags</i>

¹ *Elements of Present_Value shall be sent in order, and the first element shall be element 0 (the array size). If the notification is too large to fit within a single message, then multiple notifications shall be sent in order to convey all elements.*

[Add entry to **Table 13-2**, p. 293]

Table 13-2. Standard Objects That May Support Intrinsic Reporting

Object Type	Criteria	Event Type
<i>Global Group</i>	<i>If the IN_ALARM flag of the Member_Status_Flags property changes to a new value and remains equal to the new value for a minimum period of time specified by the Time_Delay property, and the flag for the new transition is enabled in the Event_Enable property</i>	<i>CHANGE_OF_STATUS_FLAGS</i>

[Add entry and footnote to **Table 13-3**, p. 294]

Table 13-3. Standard Object Property Values Returned in Notifications

Object	Event Type	Notification Parameters	Referenced Object's Properties
<i>Global Group</i> ⁿ	<i>CHANGE_OF_STATUS_FLAGS</i>	<i>Present_Value Referenced_Flags</i>	<i>Present_Value Member_Status_Flags</i>

¹ This parameter conveys a reference to the Log_Buffer property of the object.

ⁿ *If an event notification from a Global Group objects is too large to be accepted by the recipient, Present_Value shall be dropped from the notification.*

[Add entry to **Table 13-4**, p. 295]

Table 13-4. Notification Parameters for Standard Event Types

Event_Type	Notification Parameters	Description
<i>CHANGE_OF_STATUS_FLAGS</i>	<i>Present_Value Referenced_Flags</i>	<i>The value of the Present_Value property of the referenced object. The referenced property.</i>

[Change Clause 13.3, p.296]

13.3 Algorithmic Change Reporting

...

The following event type algorithms are specified in this standard because of their widespread occurrence in building automation and control systems. They are:

- (a) CHANGE_OF_BITSTRING
- ...
- (i) UNSIGNED_RANGE
- (j) CHANGE_OF_STATUS_FLAGS

[Add new Clause 13.3.X, p.303]

13.3.X CHANGE_OF_STATUS_FLAGS Algorithm

A CHANGE_OF_STATUS_FLAGS occurs when the referenced property, which must be of type BACnetStatusFlags, has a value of TRUE for any of its flags that also has a value of TRUE in the corresponding flag in the Selected_Flags event parameter for longer than Time_Delay seconds. For the purposes of event notification, CHANGE_OF_STATUS_FLAGS events generate a TO-OFFNORMAL transition.

After the algorithm is in the OFFNORMAL state, if the set of selected flags in the referenced property that have a value of TRUE changes, then this algorithm shall generate another TO-OFFNORMAL transition.

A CHANGE_OF_STATUS_FLAGS event clears when the referenced property has none of its flags with a value of TRUE that also has a value of TRUE in the corresponding flag in the Selected_Flags event parameter for longer than Time_Delay seconds. The clearing of a CHANGE_OF_STATUS_FLAGS generates a TO-NORMAL transition. See Figure 13-Y.

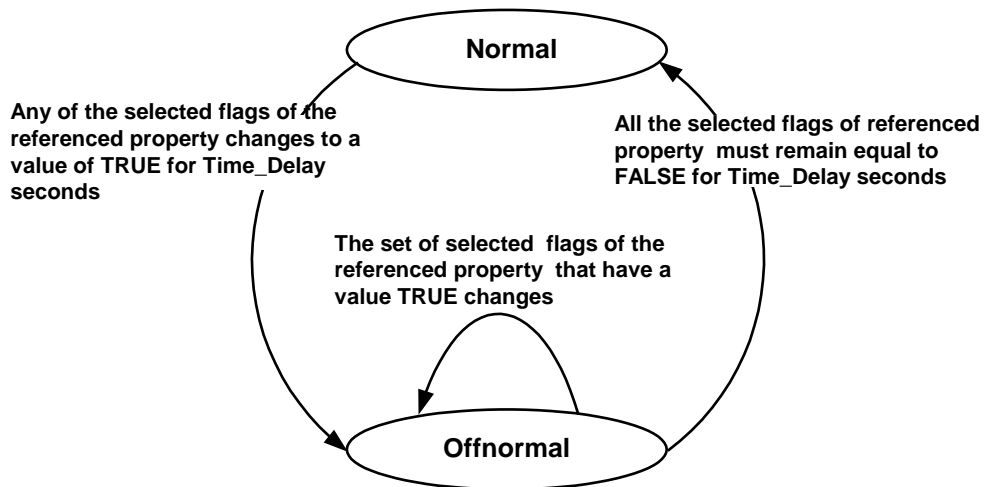


Figure 13-Y. CHANGE_OF_STATUS_FLAGS algorithm.

[Change Clause **12.12.5**, p. 188]

12.12.5 Event_Type

This property, of type BACnetEventType, indicates the type of event algorithm that is to be used to detect the occurrence of events and report to enrolled devices. This parameter is an enumerated type that may have any of the following values:

{CHANGE_OF_BITSTRING, ..., CHANGE_OF_STATUS_FLAGS }.

[Add entry to **Table 12-15**, p. 189]

Table 12-15. Event_Types, Event_States, and their Parameters

Event_Type	Event_State	Event_Parameters
CHANGE_OF_STATUS_FLAGS	NORMAL OFFNORMAL	Time_Delay Selected_Flags

[Add entry to table in Clause **12.12.7**, p. 191]

12.12.7 Event Parameters

...

Selected_Flags *This parameter, of type BACnetStatusFlags, selects which flags should be monitored for the CHANGE_OF_STATUS_FLAGS algorithm.*

[Change Clause **21**, **BACnetEventParameter** production, p. 457]

BACnetEventParameter ::= CHOICE {
 -- These choices have a one-to-one correspondence with the Event_Type enumeration with the exception of the
 -- complex-event-type, which is used for proprietary event types.

```

change-of-bitstring            [0] SEQUENCE {
                                time-delay                    [0] Unsigned,
                                bitmask                        [1] BIT STRING,
                                list-of-bitstring-values       [2] SEQUENCE OF BIT STRING
                                },
...
change-of-status-flags        [n] SEQUENCE {
                                time-delay                    [0] Unsigned,
                                selected-flags                [1] BACnetStatusFlags
                                }
    }
    
```

[Change Clause **21**, **BACnetEventType** production, p. 417]

BACnetEventType ::= ENUMERATED {
 change-of-bitstring (0),
 ...
 change-of-status-flags (n),
 }

-- Enumerated values 0-63 are reserved for definition by ASHRAE. Enumerated values
 -- 64-65535 may be used by others, subject to the procedures and constraints described
 -- in Clause 23. It is expected that these enumerated values will correspond to the use of the
 -- complex-event-type CHOICE [6] of the BACnetNotificationParameters production.
 -- ~~The last enumeration used in this version is 44.~~

[Change Clause **21**, **BACnetNotificationParameters** production, p. 462]

```

BACnetNotificationParameters::= CHOICE {
-- These choices have a one-to-one correspondence with the Event_Type enumeration with the exception of the
-- complex-event-type, which is used for proprietary event types.

    change-of-bitstring    [0] SEQUENCE {
                                referenced-bitstring    [0] BIT STRING,
                                status-flags            [1] BACnetStatusFlags
                                },
    ...
    change-of-status-flags [n] SEQUENCE {
    present-value          [0] ABSTRACT-SYNTAX.&Type OPTIONAL, -- depends on referenced property
    referenced-flags      [1] BACnetStatusFlags
    }
}

```

[Change Clause **21**, **BACnetObjectType** production, p. 463]

```

BACnetObjectType ::= ENUMERATED {
    ...
    file                (10),
    global-group        (n),
    group               (11),
    ...
    -- see global-group (n),
    ...
}
-- Enumerated values 0-127 are reserved for definition by ASHRAE.

```

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

```

BACnetObjectTypesSupported ::= BIT STRING {
    ...
    event-enrollment   (9),
    file                (10),
    group               (11),
    -- global-group     (n),
    loop                (12),
    ...
    global-group        (n),
}

```

[Add new Clause **18.3.X**, p.392]

18.3.X VALUE_NOT_INITIALIZED - An attempt was made to read a property whose value has not been initialized.

[Note to reviewers: The change to the Error production in Clause 21, adding "value-not-initialized", appears in Addendum 135-2004b -11. It was unintentionally omitted from 135-2008 but has been restored by an erratum.]

[Add to Clause **21**, new **BACnetPropertyAccessResult** production, p.471]

```

BACnetPropertyAccessResult ::= SEQUENCE {

```

```

objectIdentifier      [0]    BACnetObjectIdentifier,
propertyIdentifier    [1]    BACnetPropertyIdentifier,
propertyArrayIndex   [2]    Unsigned OPTIONAL, -- used only with array datatype
                                     -- if omitted with an array then
                                     -- the entire array is referenced

deviceIdentifier      [3]    BACnetObjectIdentifier OPTIONAL
accessResult          CHOICE {
    propertyValue      [4]    ABSTRACT-SYNTAX.&Type,
    propertyAccessError [5]    Error
}
}

```

[Add new definition to **Annex C**, p. 511]

```

...
GLOBAL-GROUP ::= SEQUENCE {
object-identifier      [75]    BACnetObjectIdentifier,
object-name            [77]    CharacterString,
object-type            [79]    BACnetObjectType,
description            [28]    CharacterString OPTIONAL,
group-members          [n]    SEQUENCE OF BACnetDeviceObjectPropertyReference,
                                     --accessed as a BACnetARRAY
group-member-names    [n+1]  SEQUENCE OF CharacterString,
                                     -- accessed as a BACnetARRAY
present-value          [85]    SEQUENCE OF BACnetPropertyAccessResult
                                     --accessed as a BACnetARRAY
event-time-stamps     [130]   SEQUENCE OF BACnetTimeStamp OPTIONAL,
                                     --accessed as a BACnetARRAY
member-status-flags   [n+2]   BACnetStatusFlags,
status-flags           [111]   BACnetStatusFlags,
event-state            [36]    BACnetEventState,
reliability            [103]   BACnetReliability OPTIONAL,
enable                 [133]   BOOLEAN,
update-interval        [118]   Unsigned OPTIONAL,
requested-update-interval [n+4] Unsigned OPTIONAL,
time-delay             [113]   Unsigned OPTIONAL,
notification-class     [17]    Unsigned OPTIONAL,
event-enable           [35]    BACnetEventTransitionBits OPTIONAL,
acked-transitions      [0]     BACnetEventTransitionBits OPTIONAL,
notify-type            [72]    BACnetNotifyType OPTIONAL,
event-time-stamps     [130]   SEQUENCE OF BACnetTimeStamp OPTIONAL,
                                     --accessed as a BACnetARRAY
notification-period    [n+3]   Unsigned,
profile-name           [167]   CharacterString OPTIONAL
}

```

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

D.X Example of a Global Group Object

The following is an example of a group object that is used to reference temperatures in a particular zone of a building.

```

Property:  Object_Identifier =      (Global Group, Instance 1)
Property:  Object_Name =           "West Wing Group"
Property:  Object_Type =           GLOBAL_GROUP
Property:  Description =           "Critical West Wing Values"

```


Property: Notification_Class = 39
Property: Event_Enable = {TRUE, TRUE, TRUE}
Property: Acked_Transitions = {TRUE, TRUE, TRUE}
Property: Notify_Type = ALARM
Property: Event_Time_Stamps = ((23-MAR-01, 18:50:21.2),
(-*-* *, *:*:* *),*
(23-MAR-01, 19:01:34.0))

Property: Notification_Period = 300

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

BACnetPropertyIdentifier ::= ENUMERATED {
...
full-duty-baseline (215),
group-members (n),
group-member-names (n+1),
high-limit (45),
...
member-of (159),
member-status-flags (n+2),
minimum-off-time (66),
...
notification-class (17), -- renamed from previous version
notification-period (n+3),
notification-threshold (137),
...
requested-shed-level (218),
requested-update-interval (n+4),
required (105),
...
-- see group-members (n),
-- see group-member-names (n+1),
-- see member-status-flags (n+2),
-- see notification-period (n+3),
-- see requested-update-interval (n+4),
}