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

ASHRAE ADDENDA

BACnet®—A Data Communication Protocol for Building Automation and Control Networks

Approved by the ASHRAE Standards Committee on June 26, 2010; by the ASHRAE Board of Directors on June 30, 2010; and by the American National Standards Institute on July 1, 2010.

This addendum was approved by a Standing Standard Project Committee (SSPC) for which the Standards Committee has established a documented program for regular publication of addenda or revisions, including procedures for timely, documented, consensus action on requests for change to any part of the standard. The change submittal form, instructions, and deadlines may be obtained in electronic form from the ASHRAE Web site (www.ashrae.org) or in paper form from the Manager of Standards.

The latest edition of an ASHRAE Standard may be purchased on the ASHRAE Web site (www.ashrae.org) or from ASHRAE Customer Service, 1791 Tullie Circle, NE, Atlanta, GA 30329-2305. E-mail: orders@ashrae.org. Fax: 404-321-5478. Telephone: 404-636-8400 (worldwide), or toll free 1-800-527-4723 (for orders in US and Canada). For reprint permission, go to www.ashrae.org/permissions.

ISSN 1041-2336

Approved American National Standard
ANSI
www.ansi.org

American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.
1791 Tullie Circle NE, Atlanta, GA 30329
www.ashrae.org

ASHRAE Standing Standard Project Committee 135
Cognizant TC: TC 1.4, Control Theory and Application
SPLS Liaison: Douglas T. Reindl

David Robin, *Chair**
Carl Neilson, *Vice-Chair*
Bernhard Isler, *Secretary**
Donald P. Alexander*
Beauford W. Atwater
David J. Branson
Barry B. Bridges*
Coleman L. Brumley, Jr.
Ernest C. Bryant
Steven T. Bushby
James F. Butler
Anthony J. Capowski
Keith A. Corbett
Jeffrey Cosiol
Troy Cowan
Harsha M. Dabholkar
Sharon E. Dinges*
Thomas Ertsgaard
Craig P. Gemmill
Daniel P. Giorgis
Ira G. Goldschmidt
David G. Holmberg
Robert L. Johnson
Stephen Karg*
Simon Lemaire
J. Damian Ljungquist*
James G. Luth
John J. Lynch
Jerald P. Martocci
Cherisse M. Nicastro
Mark A. Railsback
Carl J. Ruther
Ernest Senior
Patrick F. Sheridan
David G. Shike
Ted Sunderland
William O. Swan, III
David B. Thompson*
Daniel A. Traill
Stephen J. Treado*
J. Michael Whitcomb*
David F. White
Grant N. Wichenko

**Denotes members of voting status when the document was approved for publication.*

ASHRAE STANDARDS COMMITTEE 2009–2010

Steven T. Bushby, *Chair*
H. Michael Newman, *Vice-Chair*
Robert G. Baker
Michael F. Beda
Hoy R. Bohanon, Jr.
Kenneth W. Cooper
K. William Dean
Martin Dieryckx
Allan B. Fraser
Katherine G. Hammack
Nadar R. Jayaraman
Byron W. Jones
Jay A. Kohler
Carol E. Marriott
Merle F. McBride
Frank Myers
Janice C. Peterson
Douglas T. Reindl
Lawrence J. Schoen
Boggarm S. Setty
Bodh R. Subherwal
James R. Tauby
James K. Vallort
William F. Walter
Michael W. Woodford
Craig P. Wray
Wayne R. Reedy, *BOD ExO*
Thomas E. Watson, *CO*

Stephanie C. Reiniche, *Manager of Standards*

SPECIAL NOTE

This American National Standard (ANS) is a national voluntary consensus standard developed under the auspices of the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). *Consensus* is defined by the American National Standards Institute (ANSI), of which ASHRAE is a member and which has approved this standard as an ANS, as "substantial agreement reached by directly and materially affected interest categories. This signifies the concurrence of more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that an effort be made toward their resolution." Compliance with this standard is voluntary until and unless a legal jurisdiction makes compliance mandatory through legislation.

ASHRAE obtains consensus through participation of its national and international members, associated societies, and public review.

ASHRAE Standards are prepared by a Project Committee appointed specifically for the purpose of writing the Standard. The Project Committee Chair and Vice-Chair must be members of ASHRAE; while other committee members may or may not be ASHRAE members, all must be technically qualified in the subject area of the Standard. Every effort is made to balance the concerned interests on all Project Committees.

The Assistant Director of Technology for Standards and Special Projects of ASHRAE should be contacted for:

a. interpretation of the contents of this Standard,
b. participation in the next review of the Standard,
c. offering constructive criticism for improving the Standard, or
d. permission to reprint portions of the Standard.

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

FOREWORD

Addendum 135*p* to ANSI/ASHRAE Standard 135-2008 contains a number of changes to the current standard. 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 changes are summarized below.

135-2008*p*-1. Add a new Global Group object type, p. 2.

In the following document, language added to existing clauses of ANSI/ASHRAE 135-2008 and addenda is indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are added, plain type is used throughout.

SSPC 135 wishes to recognize the efforts of the following people in developing this addendum: Takeji Toyoda, Masaharu Nakamura, Hiroshi Ito, and Rokuro Fujii.

135-2008*p*-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 can also deliver that information in an intrinsic event notification when any of the group member objects enters a non-NORMAL state.

Addendum 135-2008*p*-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 differs from a Group object in that its members can be from anywhere in the BACnet internetwork, it supports intrinsic event reporting, and it exposes a method for sending periodic COV notifications. The Global Group object is able to monitor all referenced Status_Flags properties to detect changes to non-normal states and can initiate an event notification message conveying the values of all of the members of the group. This provides a mechanism to define a 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
COV_Resubscription_Interval	Unsigned	O
Client_COV_Increment	BACnetClientCOV	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]
COVU_Period	Unsigned	O[2]
COVU_Recipients	List of BACnetRecipient	O[2]
Profile_Name	CharacterString	O

[1] These properties are required if the object supports intrinsic reporting.
[2] These properties are required if the object sends periodic unsubscribed COV notifications for Present_Value. These properties are required to be writable if present.

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, of type BACnetARRAY of BACnetDeviceObjectPropertyReference, 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_Members and Group_Member_Names by Writing Either Property

The size of the Group_Members and Group_Member_Names properties shall be maintained so that both have the same size. If either of these arrays is writable and the size of one array is reduced, the size of the other array and the Present_Value array shall also be truncated to the new reduced size. If the size of either array is increased, the other array and the Present_Value array shall all be increased to the new expanded size and the new array elements initialized according to the requirements of each property. See Clauses 12.Y.5.2, 12.Y.6.2, and 12.Y.7.1.

12.Y.5.2 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 object or device instance numbers of the BACnetDeviceObjectPropertyReference equal to 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, of type BACnetARRAY of CharacterString, represents 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 and Group_Member_Names by Writing Either Property

See Clause 12.Y.5.1

12.Y.6.2 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 read-only property, of type BACnetARRAY of BACnetPropertyAccessResult, 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. The Present_Value may be updated based on COV notifications, polling, or a combination of the two. The method of acquisition used for any particular member is a local matter. 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. Note that the size of the Group_Members property can also be affected by changing the size of the Group_Member_Names property.

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 Present_Value is no longer tracking the group members' values, 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. In addition, the Reliability property and the corresponding state of the FAULT flag of the Status_Flags property shall be decoupled from their normal calculations when Out_Of_Service is TRUE. While the Out_Of_Service property is TRUE, the Reliability property may be changed to any value as a means of simulating specific fixed conditions or for testing purposes. Other functions that depend on the state of the Reliability property shall respond to changes made to these properties while Out_Of_Service is TRUE as if those changes had occurred by normal operation.

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 COV_Resubscription_Interval

If the Global Group is acquiring data from a remote device by COV subscription, this optional property, of type Unsigned, specifies the number of seconds between COV resubscriptions, provided that COV subscription is in effect. SubscribeCOV requests shall specify twice this lifetime for the subscription and shall specify the issuance of confirmed notifications. If COV subscriptions are in effect, the first COV subscription is issued when the Global Group object begins operation. If present, the value of this property shall be non-zero. If this property is not present, then COV subscription shall not be attempted.

12.Y.16 Client_COV_Increment

If the Global Group is acquiring COV data, this optional property, of type BACnetClientCOV, specifies the increment to be used in determining that a change of value has occurred. If all the referenced objects and properties support COV reporting according to Clause 13.1, this property may have the value NULL; in this case change of value is determined by the criteria of Clause 13.1.

12.Y.17 Time_Delay

This optional property, of type Unsigned, shall specify the minimum period of time in seconds during which the IN_ALARM flag of Member_Status_Flags must be TRUE before a TO-OFFNORMAL event is generated. It is also the minimum period of time in seconds during which the IN_ALARM flag of Member_Status_Flags must remain FALSE before a TO-NORMAL event is generated. This property is required if intrinsic reporting is supported by this object.

12.Y.18 Notification_Class

This optional 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.19 Event_Enable

This optional 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.20 Acked_Transitions

This optional 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.21 Notify_Type

This optional 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.22 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.23 COVU_Period

The optional COVU_Period property, of type Unsigned, shall indicate the amount of time in seconds between the periodic unsubscribed COV notifications performed by this object. These COV notifications convey the value of the Present_Value and Member_Status_Flags properties. If the value of COVU_Period is zero, then periodic unsubscribed COV notification messages shall not be transmitted.

12.Y.24 COVU_Recipients

This optional property, of type List of BACnetRecipient, is used to control the restrictions on which devices, if any, are to receive periodic unsubscribed COV notifications for the Present_Value. This property is required if the object sends such notifications. The value of this property shall be a list of zero or more BACnetRecipients. If the list is of length zero, a device is prohibited from sending such notifications. If the list is of length one or more, the device shall send the notifications, but only to the devices or addresses listed.

12.Y.25 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

…

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

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

BACnetReliability ::= ENUMERATED {

```
    ...
    communication-failure       (12),
    member-fault                (13),
    ...
    }
```

[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 new **Clause 13.1.X**, page 291]

13.1.X Unsubscribed COV Notifications

Some objects may share information by generating UnconfirmedCOVNotification messages without using COV subscriptions. As described in Clause 13.7, such notifications set the Subscriber Process Identifier parameter to zero to identify them as unsubscribed.

The use of UnconfirmedCOVNotification messages in this manner is not restricted, and any object can use this mechanism to distribute its properties' values to one or more recipients. The selection of which properties to send and the criteria for when to send them are sent are a local matter. A single object is not restricted to sending a single set of properties and thus may use this mechanism for different purposes with different collections of properties to different recipients.

Some standardized objects have standardized usages of this mechanism, and those are listed in Table 13-1x. Inclusion in the table does not restrict other collections of properties from being sent for other purposes.

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

Object Type	Distribution Controlled By	Criteria	Properties Reported
Device	Restart_Notification_Recipients	Device has completed the restart process. See Clause 19.3	System_Status, Time_Of_Device_Reset, Last_Restart_Reason
Global Group	COVU_Recipients	Periodic, as determined by COVU_Period	Member_Status_Flags[1], Elements of Present_Value[1]

[1] For Global Group, the elements of Present_Value shall be encoded individually each in its own BACnetPropertyValue production and shall include its array index. The elements shall be sent in index order and the first element shall be the Unsigned value at array index 0 (to inform recipients of the total array size). If the total Present_Value array is too large to fit within a single message, then multiple notifications shall be sent in order to convey all the elements. If a single element is too large to fit in a single message, it shall be encoded as an Error production with an error class of PROPERTY and an error code of VALUE_TOO_LONG. When multiple notifications are required, the index 0 element of Present_Value and the Member_Status_Flags shall be present only in the first notification.

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

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

Object Type	Criteria	Event Type
Global Group	*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*	*CHANGE_OF_STATUS_FLAGS*

[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
Global Group	*CHANGE_OF_STATUS_FLAGS*	*Present_Value*[n] *Referenced_Flags*	*Present_Value* *Member_Status_Flags*

[n] *This parameter is optional and may be omitted if it is determined to be too large. The method of such determination is a local matter.*

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

Table 13-4. Notification Parameters for Standard Event Types

Event_Type	Notification Parameters	Description
CHANGE_OF_STATUS_FLAGS	*Present_Value*[n] *Referenced_Flags*	*The value of the Present_Value property of the referenced object.* *The referenced property.*

[n] *This parameter is optional and may be omitted if it is determined to be too large. The method of such determination is a local matter.*

[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

... [items j, k, l, and m added by other addenda] ...

(n) 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.

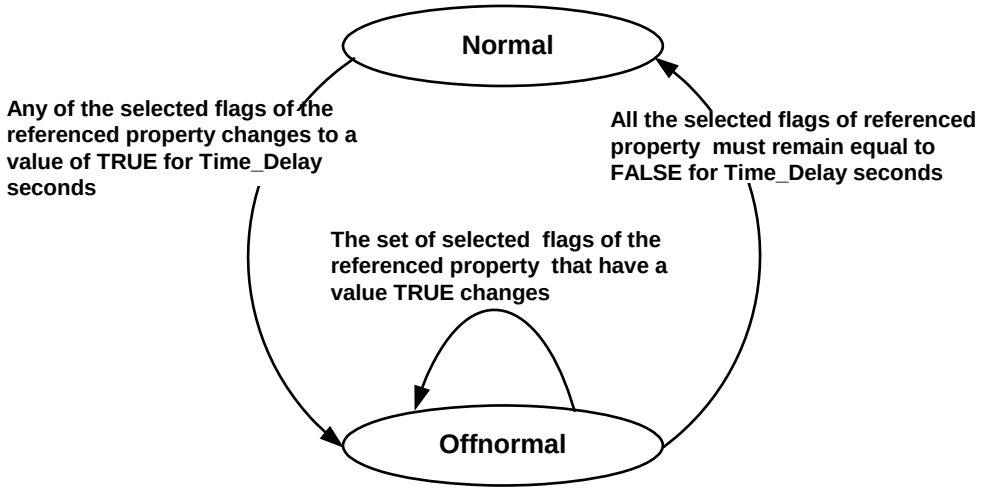

Figure 13-Y. CHANGE_OF_STATUS_FLAGS algorithm.

[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. 456]

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      [18] SEQUENCE {
                                        time-delay                  [0] Unsigned,
                                        selected-flags              [1] BACnetStatusFlags
                                        }
    }
```

[Change **Clause 21**, **BACnetEventType** production, p. 458]

BACnetEventType ::= ENUMERATED {

```
    change-of-bitstring             (0),
    ...
    change-of-status-flags          (18),
    ...
    }
```

-- 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 11.~~

[Change Clause **21**, **BACnetNotificationParameters** production, p. 461]

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      [18] 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 *(26),*
group (11),
...
-- see event-log (25),
~~-- enumeration value 26 is reserved for a future addendum~~
-- *see global-group* *(26),*
-- see trend-log-multiple (27),
...
}
-- Enumerated values 0-127 are reserved for definition by ASHRAE.

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

BACnetObjectTypesSupported ::= BIT STRING {
...
event-enrollment (9),
file (10),
group (11),
-- *global-group* *(26),*
loop (12),
...
event-log (25),
~~-- bit position 26 is reserved for a future addendum~~
global-group *(26),*
trend-log-multiple (27),
...
}

[Add new **Clauses 18.3.X1 and 18.3.X2**, p.392]

18.3.X1 VALUE_NOT_INITIALIZED - An attempt was made to read a property whose value has not been initialized.

18.3.X2 VALUE_TOO_LONG - A property value is too long to send in the current message context and an Abort is not an option, such as when sending an UnconfirmedCOVNotification.

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

[Change Clause **21**, Error production, p.443]

Error ::= SEQUENCE {
...
error-code ENUMERATED {
other (0),
value-out-of-range (37),
value-too-long *(134),*
...
-- *see value-too-long* *(134),*
...
}

[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                   [345]   SEQUENCE OF BACnetDeviceObjectPropertyReference,
                                        -- accessed as a BACnetARRAY,
group-member-names              [346]   SEQUENCE OF CharacterString, OPTIONAL,
                                        -- accessed as a BACnetARRAY,
present-value                   [85]    SEQUENCE OF BACnetPropertyAccessResult,
                                        -- accessed as a BACnetARRAY,
status-flags                    [111]   BACnetStatusFlags,
event-state                     [36]    BACnetEventState,
member-status-flags             [347]   BACnetStatusFlags,
reliability                     [103]   BACnetReliability OPTIONAL,
out-of-service                  [81]    BOOLEAN,
update-interval                 [118]   Unsigned OPTIONAL,
requested-update-interval       [348]   Unsigned OPTIONAL,
cov-resubscription-interval     [128]   Unsigned OPTIONAL,
client-cov-increment            [127]   BACnetClientCOV 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
covu-period                     [349]   Unsigned OPTIONAL,
covu-recipients                 [350]   SEQUENCE OF BACnetRecipient OPTIONAL,
profile-name                    [168]   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: Group_Members = (((Analog Input, Instance 8), Present_Value),
((Analog Input, Instance 8), Status_Flags),
((Life Safety Point, Instance 8), Present_Value),
((Life Safety Point, Instance 8), Mode),
((Life Safety Point, Instance 8), Status_Flags),
((Analog Input, Instance 9), Present_Value),
((Analog Input, Instance 9), Status_Flags),
((Life Safety Point, Instance 9), Present_Value),
((Life Safety Point, Instance 9), Mode),
((Life Safety Point, Instance 9), Status_Flags),
((Analog Input, Instance 10), Present_Value, Device, Instance 4)),
((Analog Input, Instance 10), Status_Flags, (Device, Instance 4)),
((Life Safety Point, Instance 10), Present_Value, (Device, Instance 4)),
((Life Safety Point, Instance 10), Mode, (Device, Instance 4)),
((Life Safety Point, Instance 10), Status_Flags, (Device, Instance 4)),
((Analog Input, Instance 11), Present_Value, (Device, Instance 4)),
((Analog Input, Instance 11), Status_Flags, (Device, Instance 4)),
((Life Safety Point, Instance 11), Present_Value, (Device, Instance 4)),
((Life Safety Point, Instance 11), Mode, (Device, Instance 4)),
((Life Safety Point, Instance 11), Status_Flags, (Device, Instance 4)))
Property: Group_Member_Names = ("Z8 Temp", "Z8 Temp Status", "Smoke Detector State", "Mode", "Health",
"Z9 Temp", "Z9 Temp Status", "Smoke Detector State", "Mode", "Health",
"Z10 Temp", "Z10 Temp Status", "Smoke Detector State", "Mode",
"Health","Z11 Temp", "Z11 Temp Status", "Smoke Detector State", "Mode",
"Health")
Property: Present_Value = (((Analog Input, Instance 8), Present_Value, 69.7),
((Analog Input, Instance 8),
Status_Flags,{FALSE, FALSE, FALSE,FALSE}),
((Life Safety Point, Instance 8), Present_Value, QUIET)
((Life Safety Point, Instance 8), Mode, ON),
((Life Safety Point, Instance 8),
Status_Flags, {FALSE, FALSE, FALSE,FALSE}),
((Analog Input, Instance 9), Present_Value, 71.2),
((Analog Input, Instance 9),
Status_Flags,{FALSE, FALSE, FALSE,FALSE}),
((Life Safety Point, Instance 9), Present_Value, QUIET),
((Life Safety Point, Instance 9), Mode, ON),
((Life Safety Point, Instance 9),
Status_Flags, {FALSE, FALSE, FALSE,FALSE}),
((Device, Instance 4), (Analog Input, Instance 10), Present_Value, -50),
((Device, Instance 4), (Analog Input, Instance 10),
Status_Flags, {TRUE,TRUE,FALSE,FALSE}),
((Device, Instance 4), (Life Safety Point, Instance 10), Present_Value,QUIET),
((Device, Instance 4), (Life Safety Point, Instance 10), Mode, ON),
((Device, Instance 4), (Life Safety Point, Instance 10),
Status_Flags, {FALSE, FALSE, FALSE, FALSE}),
((Device, Instance 4) , (Analog Input, Instance 11), Present_Value, 69.7),
((Device, Instance 4), (Analog Input, Instance 11),
Status_Flags, {FALSE,FALSE, FALSE, FALSE}),
((Device, Instance 4), (Life Safety Point, Instance 11), Present_Value,QUIET),

```
                                    ((Device, Instance 4), (Life Safety Point, Instance 11), Mode, ON),
                                    ((Device, Instance 4), (Life Safety Point, Instance 11)
                                            Status_Flags,{FALSE, FALSE, FALSE, FALSE}) )
Property:   Status_Flags =          {TRUE, TRUE, FALSE, FALSE}
Property:   Event_State =           FAULT
Property:   Member_Status_Flags =   {TRUE, TRUE, FALSE, FALSE}
Property:   Reliability =           UNRELIABLE_OTHER
Property:   Out_Of_Service -=       FALSE
Property:   Update_Interval =       10
Property:   Requested_Update_Interval =10
Property:   Time_Delay =            10
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:   COVU_Period =           300
Property:   COVU_Recipients =       ((0,X''))     -- local broadcast address
```

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

```
BACnetPropertyIdentifier ::= ENUMERATED {
    ...
    cov-resubscription-interval         (128),
    covu-period                         (349),
    covu-recipients                     (350),
    --  current-notify-time             (129),    This property was deleted in version 1 revision 3.
    ...
    full-duty-baseline                  (215),
    group-members                       (345),
    group-member-names                  (346),
    high-limit                          (45),
    ...
    member-of                           (159),
    member-status-flags                 (347),
    minimum-off-time                    (66),
    ...
    requested-shed-level                (218),
    requested-update-interval           (348),
    required                            (105),
    ...
    ...
    -- see group-members                (345),
    -- see group-member-names           (346),
    -- see member-status-flags          (347),
    -- see requested-update-interval    (348),
    -- see covu-period                  (349),
    -- see covu-recipients              (350),
}
```

[Add a new entry to **History of Revisions**, p. 688]

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

HISTORY OF REVISIONS

Protocol		***Summary of Changes to the Standard***
Version	***Revision***	
…	…	**…**
1	11	**Addendum *p* to ANSI/ASHRAE 135-2008** Approved by the ASHRAE Standards Committee June 26, 2010; by the ASHRAE Board of Directors June 30, 2010; and by the American National Standards Institute July 1, 2010. 1. Add a new Global Group object type.

POLICY STATEMENT DEFINING ASHRAE'S CONCERN FOR THE ENVIRONMENTAL IMPACT OF ITS ACTIVITIES

ASHRAE is concerned with the impact of its members' activities on both the indoor and outdoor environment. ASHRAE's members will strive to minimize any possible deleterious effect on the indoor and outdoor environment of the systems and components in their responsibility while maximizing the beneficial effects these systems provide, consistent with accepted standards and the practical state of the art.

ASHRAE's short-range goal is to ensure that the systems and components within its scope do not impact the indoor and outdoor environment to a greater extent than specified by the standards and guidelines as established by itself and other responsible bodies.

As an ongoing goal, ASHRAE will, through its Standards Committee and extensive technical committee structure, continue to generate up-to-date standards and guidelines where appropriate and adopt, recommend, and promote those new and revised standards developed by other responsible organizations.

Through its *Handbook*, appropriate chapters will contain up-to-date standards and design considerations as the material is systematically revised.

ASHRAE will take the lead with respect to dissemination of environmental information of its primary interest and will seek out and disseminate information from other responsible organizations that is pertinent, as guides to updating standards and guidelines.

The effects of the design and selection of equipment and systems will be considered within the scope of the system's intended use and expected misuse. The disposal of hazardous materials, if any, will also be considered.

ASHRAE's primary concern for environmental impact will be at the site where equipment within ASHRAE's scope operates. However, energy source selection and the possible environmental impact due to the energy source and energy transportation will be considered where possible. Recommendations concerning energy source selection should be made by its members.