ANSI/ASHRAE Addendum *a* to
ANSI/ASHRAE Standard 135-2001

ASHRAE® STANDARD

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

Approved by the ASHRAE Standards Committee on October 5, 2003; by the ASHRAE Board of Directors on January 29, 2004; and by the American National Standards Institute on February 25, 2004.

This standard is under continuous maintenance 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, *http://www.ashrae.org*, or in paper form from the Manager of Standards. The latest edition of an ASHRAE Standard and printed copies of a public review draft may be purchased 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 U.S. and Canada).

ISSN 1041-2336

Approved American National Standard
ANSI

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

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 members may or may not be members of ASHRAE, 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 Manager of Standards 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,
d. permission to reprint portions of the Standard.

(This foreword is not part of this 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 a number of independent substantive changes to the BACnet standard. These modifications are the result of change proposals made pursuant to the continuous maintenance procedures contained in the *Manual for Processing ASHRAE Standards* and *PC Guidance* and of deliberations within Standing Standard Project Committee 135. The proposed changes are summarized below.

135*a*-1. Add Partial Day Scheduling to the Schedule object, p. 1.

135*a*-2. Enable reporting of proprietary events by the Event Enrollment object, p. 9.

135*a*-3. Allow detailed error reporting when all ReadPropertyMultiple accesses fail, p. 11.

135*a*-4. Remove the Recipient property from the Event Enrollment object, p. 12.

135*a*-5. Add the capability to issue I-Am responses on behalf of MS/TP slave devices, p.15.

135*a*-6. Add a new silenced mode to the DeviceCommunicationControl service, p. 19.

135*a*-7. Add 21 new engineering units, p. 20.

135*a*-8. Specify the behavior of a BACnetArray when its size is changed, p. 22.

135*a*-9. Clarify the behavior of a BACnet router when it receives an unknown network message type, p.25.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2001 and Addenda is indicated through the use of *italics* while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are to be added, plain type is used throughout.

135*a*-1. Add Partial Day Scheduling to the Schedule object.

Addendum 135*a*-1

[Add to **Clause 12** introduction, p. 129]

…

MULTI_STATE_FAULT	The Present_Value of the Multi-state object is equal to one of the states in the Fault_Values property and no other fault has been detected.
CONFIGURATION_ERROR	*The object's properties are not in a consistent state.*

[Change **Clause 12.22**, p.224]

12.22 Schedule Object Type

The Schedule object type defines a standardized object used to describe a periodic schedule that may recur during a range of dates, with optional exceptions ~~on arbitrary dates~~ *at arbitrary times on arbitrary dates*. The Schedule object also serves as a binding between these scheduled times and the writing of specified "values" to specific properties of specific objects at those times. The Schedule object type and its properties are summarized in Table 12-26 and described in detail in this subclause.

Schedules are divided into days, of which there are two types: normal days within a week and exception days. *Both types of days can specify scheduling events for either the full day or portions of a day, and a priority mechanism defines which scheduled event is in control at any given time.*

The current state of the Schedule object is represented by the value of its Present_Value property, which is normally calculated using the time/value pairs from the Weekly_Schedule and Exception_Schedule properties, with a default value for use when no schedules are in effect. Details of this calculation are provided in 12.22.4.

~~It is assumed that the scheduler will exhibit restorative behavior in the event that the BACnet Device containing the schedule is restarted or the time is changed in the BACnet Device. The model for restoration assumes that each day's schedule is circular in nature. Thus, if the BACnet Device is restarted after midnight but prior to the first time in the list of BACnetTimeValues for that day, then the last value on the list for that day is used as the restoration value. If some other value is desired, then an explicit time of 00:00 shall be the first entry in the list.~~

Versions of the Schedule object prior to Protocol_Revision 4 only support schedules that define an entire day, from midnight to midnight. For compatibility with these versions, this whole day behavior can be achieved by using a specific schedule format. Weekly_Schedule and Exception_Schedule values that begin at 00:00, and do not use any NULL values, will define schedules for the entire day. Property values in this format will produce the same results in all versions of the Schedule object.

Table 12-26. Properties of the Schedule Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	Any	R
Description	CharacterString	O
Effective_Period	BACnetDateRange	R
Weekly_Schedule	BACnetARRAY[7] of BACnetDailySchedule	O[1]
Exception_Schedule	BACnetARRAY[N] of BACnetSpecialEvent	O[1]
Schedule_Default	*Any*	*R*
List_Of_Object_Property_References	List of BACnetDeviceObjectPropertyReference	R
Priority_For_Writing	Unsigned (1..16)	R
Status_Flags	*BACnetStatusFlags*	*R*
Reliability	*BACnetReliability*	*R*
Out_Of_Service	*BOOLEAN*	*R*
Profile_Name	CharacterString	O

[1] At least one of these properties is required.

[Change **Clause 12.22.4**, p.224]

12.22.4 Present_Value

This property indicates the current value of the schedule*,* ~~i.e. the value most recently written to a referenced property of one member of List_Of_Object_Property_References.~~ ~~This~~ *which* may be any primitive datatype. As a result, *most* analog, binary and enumerated values may be scheduled. ~~If the List_Of_Object_Property_References is empty, then the value of this property will be that which would have been most recently written to the List_Of_Object_Property_References.~~ *This property shall be writable when Out_Of_Service is TRUE (see 12.22.11).*

Any change in the value of this property shall be written to all members of the List_Of_Object_Property_References property. An error writing to any member of the list shall not stop the Schedule object from writing to the remaining members.

The normal calculation of the value of the Present_Value property is illustrated as follows (the actual algorithm used is a local matter but must yield the same results as this one):

1. *Find the highest relative priority (as defined by 12.22.8) Exception_Schedule array element that is in effect for the current day and whose current value (see method below) is not NULL, and assign that value to the Present_Value property.*
2. *If the Present_Value was not assigned in the previous step, then evaluate the current value of the Weekly_Schedule array element for the current day and if that value is not NULL, assign it to the Present_Value property.*
3. *If the Present_Value was not assigned in the previous steps, then assign the value of the Schedule_Default property to the Present_Value property.*

The method for evaluating the current value of a schedule (either exception or weekly) is to find the latest element in the list of BACnetTimeValues that occurs on or before the current time, and then use that element's value as the current value for the schedule. If no such element is found, then the current value for the schedule shall be NULL.

These calculations are such that they can be performed at any time and the correct value of Present_Value property will result. These calculations must be performed at 00:00 each day, whenever the device resets, whenever properties that can affect the results are changed, whenever the time in the device changes by an amount that may

have an effect on the calculation result, and at other times, as required, to maintain the correct value of the Present_Value property through the normal passage of time.

Note that the Present_Value property will be assigned the value of the Schedule_Default property at 00:00 of any given day, unless there is an entry for 00:00 in effect for that day. If a scheduled event logically begins on one day and ends on another, an entry at 00:00 shall be placed in the schedule that is in effect for the second day, and for any subsequent days of the event's duration, to ensure the correct result whenever Present_Value is calculated.

[Change **Clause 12.22.6**, p.224]

12.22.6 Effective_Period

This property specifies the range of dates within which the Schedule object is active. Seasonal scheduling may be achieved by defining several SCHEDULE objects with non-overlapping Effective_Periods to control the same property references. *Upon entering its effective period, the object shall calculate its Present_Value and write that value to all members of the List_Of_Object_Property_References property. An error writing to any member of the list shall not stop the Schedule object from writing to the remaining members.*

[Change **Clause 12.22.7**, p.225]

12.22.7 Weekly_Schedule

This property is a BACnetARRAY containing exactly seven elements. Each of the elements 1-7 contains a BACnetDailySchedule. A BACnetDailySchedule consists of a list of BACnetTimeValues that are (time, value) pairs, which describe the sequence of schedule actions on one day of the week when no Exception_Schedule is in effect. The array elements 1-7 correspond to the days Monday - Sunday, respectively. The Weekly_Schedule is an optional property, but either the Weekly_Schedule or a non-empty Exception_Schedule shall be supported in every instance of a Schedule object.

If the Weekly_Schedule property is written with a schedule item containing a datatype not supported by this instance of the Schedule object (e.g., the List_Of_Object_Property_References property cannot be configured to reference a property of the unsupported datatype), the device may return a Result(-) response, specifying an 'Error Class' of PROPERTY and an 'Error Code' of DATATYPE_NOT_SUPPORTED.

[Change **Clause 12.22.8**, p.225]

12.22.8 Exception_Schedule

This property is a BACnetARRAY of BACnetSpecialEvents. Each BACnetSpecialEvent describes a sequence of schedule actions that takes precedence over the normal day's behavior on a specific day or days.

BACnetSpecialEvent ::= (Period, List of BACnetTimeValue, EventPriority)

Period ::= Choice of {BACnetCalendarEntry | CalendarReference}

EventPriority ::= Unsigned (1..16)

The Period may be a BACnetCalendarEntry or it may refer to a Calendar object as described in 12.8. A BACnetCalendarEntry would be used if the Exception_Schedule is specific to this Schedule object, while calendars might be defined for common holidays to be referenced by multiple Schedule objects. Each BACnetCalendarEntry is either an individual date (Date), range of dates (BACnetDateRange), or month/week-of-month/day-of-week specification (BACnetWeekNDay). If the current date matches any of the calendar entry criteria, the Exception Schedule would be activated and the list of BACnetTimeValues would be enabled for use.

Individual fields of the various constructs of the BACnetCalendarEntry may also have a "wildcard" value used for determining whether the current date falls within the Period of the Exception Schedule. In a date range, for example,

if the startDate is a wildcard, it means "any date up to and including the endDate." If the endDate is a wildcard, it means "any date from the startDate on." If the calendar entry were a BACnetWeekNDay with wildcard for month and week-of-month fields but with a specific day-of-week, it would mean that the Exception Schedule would apply on that day-of-week all year long.

Each BACnetSpecialEvent contains an EventPriority that determines its importance relative to other BACnetSpecialEvents within the same Exception_Schedule. Since SpecialEvents within the same Exception_Schedule may have overlapping periods, it is necessary to have a mechanism to determine ~~which SpecialEvent applies~~ *the relative priorities for the SpecialEvents that apply* on any given day. If more than one SpecialEvent applies to a given day, the *relative priority of the SpecialEvents shall be determined by their* ~~SpecialEvent with the highest~~ EventPriority *values* ~~shall have higher priority for evaluation take effect and all other applicable SpecialEvents shall be ignored for that day~~. If multiple overlapping SpecialEvents have the same ~~priority,~~ *EventPriority value*, then the SpecialEvent ~~occurring earliest~~ *with the lowest index number* in the array shall ~~take effect for that day~~ *have higher relative priority*. The highest EventPriority is 1 and the lowest is 16. The EventPriority is not related to the Priority_For_Writing property of the Schedule object.

If a BACnet Device supports writing to the Exception_Schedule property, all possible choices in the BACnetSpecialEvents shall be supported.

If the Exception_Schedule property is written with a schedule item containing a datatype not supported by this instance of the Schedule object (e.g., the List_Of_Object_Property_References property cannot be configured to reference a property of the unsupported datatype), the device may return a Result(-) response, specifying an 'Error Class' of PROPERTY and an 'Error Code' of DATATYPE_NOT_SUPPORTED.

[Renumber **Clause 12.22.9**, List_Of_Object_Property_References, to **12.22.10;**
Renumber **Clause 12.22.10**, Priority_For_Writing, to **12.22.11;**
Renumber **Clause 12.22.11**, Profile_Name, to **12.22.15**]

[Insert new **Clause 12.22.9**, p.225]

12.22.9 Schedule_Default

This property holds a default value to be used for the Present_Value property when no other scheduled value is in effect (see 12.22.4). This may be any primitive datatype.
If the Schedule_Default property is written with a value containing a datatype not supported by this instance of the Schedule object (e.g., the List_Of_Object_Property_References property cannot be configured to reference a property of the unsupported datatype), the device may return a Result(-) response, specifying an 'Error Class' of PROPERTY and an 'Error Code' of DATATYPE_NOT_SUPPORTED.

[Insert new **Clause 12.22.12**, p.226]

12.22.12 Status_Flags

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

{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}

where:

IN_ALARM	The value of this flag shall be logical FALSE (0).
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 schedule object has been overridden by some mechanism local to the BACnet Device. In this context "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. 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).

[Insert new **Clause 12.22.13**, p.226]

12.22.13 Reliability

The Reliability property, of type BACnetReliability, provides an indication that the properties of the schedule object are in a consistent state. All non-NULL values used in the Weekly_Schedule, the Exception_Schedule, and the Schedule_Default properties shall be of the same datatype, and all members of the List_Of_Object_Property_References shall be writable with that datatype. If these conditions are not met, then this property shall have the value CONFIGURATION_ERROR. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, CONFIGURATION_ERROR, UNRELIABLE_OTHER}.

If the List_Of_Object_Property_References contains a member that references a property in a remote device, the detection of a configuration error may be delayed until an attempt is made to write a scheduled value.

[Insert new **Clause 12.22.14**, p.226]

12.22.14 Out_Of_Service

The Out_Of_Service property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the internal calculations of the schedule object are used to determine the value of the Present_Value property. This means that the Present_Value property is decoupled from the internal calculations and will not track changes to other properties when Out_Of_Service is TRUE. Other functions that depend on the state of the Present_Value, such as writing to the members of the List_Of_Object_Property_References, shall respond to changes made to that property while Out_Of_Service is TRUE, as if those changes had occurred by internal calculations.

[Renumber **Clauses 18.3.2** through **18.3.11** to **18.3.3** through **18.3.12**, and insert new **Clause 18.3.2**, p.334]

18.3.2 DATATYPE_NOT_SUPPORTED - The data is of, or contains, a datatype not supported by this instance of this property.

[Change **Clause 21,** p.385]

```
Error     ::= SEQUENCE {
…
     error-code     ENUMERATED {
                        other                                   (0),
                        authentication-failed                   (1),
                        character-set-not-supported             (41),
                        configuration-in-progress               (2),
                        datatype-not-supported                  (47),
                        device-busy                             (3),
                        …
                        .
               -- see   optional-functionality-not-supported,   (45),
               -- see   invalid-configuration-data               (46),
               -- see   datatype-not-supported                  (47),
                        ...
```

```
        }
        -- Enumerated values 0-255 are reserved for definition by ASHRAE. Enumerated values
        -- 256-65535 may be used by others subject to the procedures and constraints described
        -- in Clause 23. The last enumeration used in this version is ~~46~~ 47.
    }
```

[Change **Clause 21**, p. 387]

```
    Date ::= [APPLICATION 10] OCTET STRING (SIZE(4)) -- see 20.2.12
    --  first octet     year minus 1900      X'FF' = unspecified
    --  second octet    month (1..~~12~~14)      1 = January
    --                                       odd months=13
    --                                       even months=14
    --                                       X'FF' = unspecified
    --  third octet     day of month (1..~~31~~32),
    --                                           32=last day of month
    --                                       X'FF' = unspecified
    --  fourth octet    day of week  (1..7)  1 = Monday
    --                                       7 = Sunday
    --                                       X'FF' = unspecified
```

[Change **Clause 21**, pp. 399-403]

```
    BACnetPropertyIdentifier ::= ENUMERATED {
        …
        resolution                  (106),
        schedule-default            (174),
        segmentation-supported      (107),
        …
        -- see profile-name        (168),
        -- see schedule-default    (174),
        …
                }
```

[Change **Clause 21**, p. 405]

```
    BACnetReliability ::= ENUMERATED {
        …
        multi-state-fault       (9),
        configuration-error     (10),
        ...
            }
```

[Change **Clause 21**, p. 407]

```
    BACnetWeekNDay ::= OCTET STRING (SIZE (3))
    --  first octet     month (1..~~12~~14)   January = 1,
    --                                       odd months=13
    --                                       even months=14
    --                                       X'FF' = any month
    --  second octet    weekOfMonth      where: 1 = days numbered 1-7
    --                                       2 = days numbered 8-14
    --                                       3 = days numbered 15-21
    --                                       4 = days numbered 22-28
    --                                       5 = days numbered 29-31
    --                                       6 = last 7 days of this month
```

```
--                                          X'FF' = any week of this month
--   third octet     dayOfWeek (1..7) where   1 = Monday
--                                            7 = Sunday
--                                            X'FF' = any day of week
```

[Change **Annex C**, p. 437]

```
SCHEDULE ::= SEQUENCE {
…
    exception-schedule                  [38]  SEQUENCE OF BACnetSpecialEvent OPTIONAL,
                                              -- accessed as a BACnetARRAY
    schedule-default                    [174] ABSTRACT-SYNTAX.&Type, -- Any primitive datatype,
    list-of-object-property-references  [54]  SEQUENCE OF BACnetDeviceObjectPropertyReference,
    priority-for-writing                [88]  Unsigned (1..16),
    status-flags                        [111] BACnetStatusFlags,
    reliability                         [103] BACnetReliability,
    out-of-service                      [81]  BOOLEAN,
    profile-name                        [168] CharacterString OPTIONAL
    }
```

[Change **Annex D.22**, p. 455]

The following is an example of a Schedule object that is used to control a classroom during the school year. In this example, a different Schedule object is assumed to be defined for the remainder of the calendar year. The reference property of this schedule is the Present_Value of a Binary Output object that controls a rooftop unit providing conditioned air to room 208.

```
…
Property: Weekly_Schedule =             {((8:00,ACTIVE),(17:00,INACTIVE)),
                                         ((8:00,ACTIVE)),
                                         ((8:00,ACTIVE),(17:00,INACTIVE)),
                                         ((8:00,ACTIVE),(17:00,INACTIVE),(19:00,ACTIVE),
                                            (23:30,INACTIVE)),
                                         ((8:00,ACTIVE),(17:00,INACTIVE)),
                                         ((00:00,INACTIVE)),
                                         ((10:00,ACTIVE),(17:00,INACTIVE))}
Property: Exception_Schedule =          {((23-NOV-1995),(0:00,INACTIVE),10),
                                         ((HOLIDAYS,(0:00,INACTIVE),11),
                                         ((5-MAR-1996)-(7-MAR-1996),((0:00,INACTIVE),(9:00,ACTIVE),
                                                (14:00,INACTIVE)),6)
                                         ((8-MAR-1996),((10:00,INACTIVE),(11:00,NULL)),7)}
Property: List_Of_Object_Property_References =      ((Binary Output, Instance 9), Present_Value)
Property: Priority_For_Writing =        15
Property: Status_Flags =                {FALSE, FALSE, FALSE, FALSE}
Property: Reliability =                 NO_FAULT_DETECTED
Property: Schedule_Default =            INACTIVE
Property: Out_Of_Service =              FALSE
```

On normal Mondays, Wednesdays, and Fridays the rooftop is made ACTIVE at 8 A.M. and INACTIVE at 5 P.M. On normal Tuesdays, the rooftop is made ACTIVE at 8 A.M. and is not otherwise affected by the schedule. On normal Thursdays, the rooftop is made ACTIVE at 8 A.M., INACTIVE at 5 P.M., ACTIVE again at 7 P.M., and INACTIVE at 11:30 P.M. *On Saturdays the rooftop is INACTIVE all day. On Sundays the rooftop is made ACTIVE at 10 A.M. and INACTIVE at 5 P.M.*

On ~~Friday~~ *Thursday*, November 23, 1995, the normal ~~Friday~~ *Thursday* schedule is overridden by the first SPECIALEVENT in the Exception_Schedule list, which sets the rooftop INACTIVE for the entire day.

On Monday, February 19, 1996, the normal Monday schedule is overridden by the second SPECIALEVENT in the Exception_Schedule list, which follows the HOLIDAYS calendar described in D.8. Since February 19, 1996, Presidents' Day, is in the HOLIDAYS object's DateList, the TIMEVALUEs associated with the second SPECIALEVENT are followed, which set the rooftop INACTIVE for the entire day.

The third SPECIALEVENT overrides the normal schedule between March 5, 1996, and March 7, 1996, for a teachers' conference. On these three days, the rooftop is made ACTIVE at 9 A.M. and INACTIVE at 2 P.M.

The fourth SPECIALEVENT overrides the normal Friday schedule on March 8, 1996, to create an hour of inactivity in the middle of the day for maintenance. On this day, the rooftop is made ACTIVE at 8 A.M. (by the normal schedule), INACTIVE at 10 A.M. (by the exception schedule), ACTIVE again at 11 A.M. (by the exception relinquishing, and the normal schedule resuming) and INACTIVE at 5 P.M. (by the normal schedule).

135*a*-2. Enable reporting of proprietary events by the Event Enrollment object.

Addendum 135*a*-2

[Change **Clause 12.11.5**, p. 176]

12.11.5 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_STATE, CHANGE_OF_VALUE, COMMAND_FAILURE, FLOATING_LIMIT, OUT_OF_RANGE, BUFFER_READY, ~~CHANGE_OF_LIFE_SAFETY}~~
CHANGE_OF_LIFE_SAFETY, EXTENDED}
…

[Change **Clause 13.3,** para. 5, p. 239]

13.3 Algorithmic Change Reporting

…

Event enrollment objects shall use standard event types for reporting the values of parameters relevant to the event. These values are returned in the `Event Values' parameter of the ConfirmedEventNotification and UnconfirmedEventNotification services. The event type determines which set of values to return for each of the standard event types. These return values are summarized in Table 13-4. ~~Proprietary event types shall not be used with Event Enrollment objects because they would be forced to return a list of property values without being able to specify the object type of the properties returned.~~ *Event Enrollment objects must report proprietary event algorithms using the extended-event notification type.*

[Add new production in **Clause 21**, p. 389]

BACnetDeviceObjectPropertyValue *::= SEQUENCE {*

deviceIdentifier	*[0] BACnetObjectIdentifier,*
objectIdentifier	*[1] BACnetObjectIdentifier,*
propertyIdentifier	*[2] BACnetPropertyIdentifier,*
arrayIndex	*[3] Unsigned – OPTIONAL,*
value	*[4] ABSTRACT-SYNTAX.&Type*

}

[Change BACnetEventParameter production in **Clause 21**, p. 393]

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-life-safety	[8] SEQUENCE {	
	time-delay	[0] Unsigned,
	list-of-life-safety-alarm-values	[1] SEQUENCE OF BACnetLifeSafetyState,
	list-of-alarm-values	[2] SEQUENCE OF BACnetLifeSafetyState,
	mode-property-reference	[3] BACnetDeviceObjectPropertyReference
	~~}~~ *},*	

```
        extended            [9] SEQUENCE {
                                vendorId            [0] Unsigned,
                                extendedEventType   [1] Unsigned,
                                parameters          [2] SEQUENCE OF CHOICE {
                                                        null        NULL,
                                                        real        REAL,
                                                        integer     Unsigned,
                                                        boolean     BOOLEAN,
                                                        double      DOUBLE,
                                                        octet       OCTET STRING,
                                                        bitstring   BIT STRING,
                                                        enum        ENUMERATED,
                                                        reference   [0] BACnetDeviceObjectProperty
                                                        }
                                }
        }
```

[Change BACnetEventType production **in Clause 21**, p. 394]

BACnetEventType ::= ENUMERATED {
…
buffer-ready (7),
change-of-life-safety ~~(8)~~ *(8),*
extended (9)
}
-- 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 ~~8.~~ *9.*

[Change BACnetNotificationParameters production in **Clause 21**, p. 397]

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

```
        …
                                    ~~}~~ },
        extended        [9] SEQUENCE {
                            vendorId            [0] Unsigned,
                            extendedEventType   [1] Unsigned,
                            parameters          [2] SEQUENCE OF CHOICE {
                                                    null            NULL,
                                                    real            REAL,
                                                    integer         Unsigned,
                                                    boolean         BOOLEAN,
                                                    double          DOUBLE,
                                                    octet           OCTET STRING,
                                                    bitstring       BIT STRING,
                                                    enum            ENUMERATED,
                                                    propertyValue   [0] BACnetDeviceObjectPropertyValue
                                                    }
                            }
    }
```

135*a*-3. Allow detailed error reporting when all ReadPropertyMultiple accesses fail.

Addendum 135*a*-3

[Change **Clause 15.7.2**, p. 292]

15.7.2 Service Procedure

After verifying the validity of the request, the responding BACnet-user shall attempt to access the specified properties of the specified objects and shall construct a `List of Read Access Results' in the order specified in the request. If the `List of Property References' portion of the `List of Read Access Specifications' parameter contains the property identifier ALL, REQUIRED, or OPTIONAL, then the `List of Read Access Results' shall be constructed as if each property being returned had been explicitly referenced (see 15.7.3.1.2). While there is no requirement that the request be carried out "atomically," nonetheless the responding BACnet-user shall ensure that all readings are taken in the shortest possible time subject only to higher priority processing. The request shall continue to be executed until an attempt has been made to access all specified properties. If none of the specified objects is found or if none of the specified properties of the specified objects can be accessed, *either* a `Result(-)' primitive *or a `Result(+)' primitive that returns error codes for all properties* shall be issued. If any of the specified properties of the specified objects can be accessed, then a `Result(+)' primitive shall be issued, which returns all accessed values and error codes for all properties that could not be accessed.

135*a*-4. Remove the Recipient property from the Event Enrollment object.

Addendum 135*a*-4

[Change **Table 12-13**, p. 175]

Table 12-13. Properties of the Event Enrollment Object Type

Property Identifier	Property Datatype	Conformance Code
…	…	…
Notification_Class	Unsigned32	~~O~~[1] *R*
~~Recipient~~	~~BACnetRecipient~~	~~O~~[2]
~~Process_Identifier~~	~~Unsigned~~	~~O~~[2]
~~Priority~~	~~Unsigned~~	~~O~~[2]
~~Issue_Confirmed_Notifications~~	~~BOOLEAN~~	~~O~~[2]
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	O
…	…	…

~~[1] The Notification_Class property shall be present if the Recipient property is not.~~

~~[2] All of these properties shall be present if the Notification_Class property is not.~~

[Change **Clause 12.11.12**, p. 178]

12.11.12 Notification_Class

This property, of type Unsigned, implicitly references a Notification Class object in the device containing the Event Enrollment object. The class is used to specify the handling, reporting, and acknowledgment characteristics for one or more event-initiating objects. ~~If the Notification_Class property is not present, then the Recipient, Process_Identifier, Priority, and Issue_Confirmed_Notifications properties shall be present. If the Notification_Class property is present but the Recipient property has a non-NULL value (meaning that the Recipient property is being used for notification), then the Notification_Class property may contain any value and shall not be used.~~

[Delete **Clauses 12.11.13, 12.11.14, 12.11.15, 12.11.16**, pp. 178-179 and renumber **Clauses 12.11.17** to **12.11.13**, and **12.11.18** to **12.11.14**.]

[Change **Clause 13**, para. 5, p 233]

13. ALARM AND EVENT SERVICES

…

Intrinsic and algorithmic change reporting rely on the concept of event classification for selective reporting of individual occurrences of events. Particular intrinsic or algorithmic change events ~~may~~ specify a notification class number that is directly related to a Notification Class object within the initiating device. The Notification Class object is used to specify the handling and routing of events to one or more destinations. The Notification Class object defines the priorities to be used in event-notification messages, whether acknowledgment by an application process or human operator is required, and at what time periods during the week given destinations are to be used. ~~Algorithmic change reporting may specify a BACnetRecipient rather than a Notification_Class.~~

[Change **Clause 13.3**, para. 1, p. 239]

13.3 Algorithmic Change Reporting

Algorithmic change reporting enables a BACnet device to provide one or more alarm or event sources, defined by Event Enrollment objects, to generate alarm or event notifications that may be directed to one or more destinations. Any of the standardized algorithms may be used to establish criteria for change reporting. Once established, occurrences of change may be reported to one or more destinations based on further criteria. Changes of value of specific properties of an object may be programmed to trigger event notifications to be sent to one or more destinations based on notification class. ~~Alternatively, a given algorithmic change may notify a single recipient.~~ Typically, event notifications are sent to application programs represented by processes within a notification-client device. The ConfirmedEventNotification and UnconfirmedEventNotification services are used by the notification-server to convey notifications.

[Change **Clause 13.11.1.2**, para. 1, p. 260]

13.11.1.1.2 Enrollment Filter

This parameter, of type BACnetRecipientProcess, shall provide a means of restricting the set of objects that are to be summarized. Only those objects for which the specified BACnetRecipient and Process Identifier are enrolled to receive notifications, either confirmed or unconfirmed, shall be summarized. In this case, "enrolled" shall mean that ~~either~~ *an event-initiating object references a Notification Class object containing one or more BACnetDestinations containing the indicated Process Identifier and BACnetRecipient.*

~~(a) an Event Enrollment object contains the indicated Process Identifier and BACnetRecipient or~~
~~(b) an event-initiating object references a Notification Class object containing one or more BACnetDestinations containing the indicated Process Identifier and BACnetRecipient.~~

If this parameter is omitted, it shall mean that event-initiating objects shall be summarized without regard to enrollment status.

[Change BACnetPropertyIdentifier production in **Clause 21**, p. 401]

BACnetPropertyIdentifier ::= ENUMERATED {
    acked-transitions (0),
    …
~~issue-confirmed-notifications (51),~~
-- issue-confirmed-notifications-- (51), This property was deleted in version 1 revision 4
    …
~~recipient (101),~~
-- recipient (101), This property was deleted in version 1 revision 4
    …
    }

[Change **ANNEX C**, p.433]

EVENT-ENROLLMENT ::= SEQUENCE {
    …
    notification-class [17] Unsigned32 ~~OPTIONAL~~
    ~~recipient [101] BACnetRecipient OPTIONAL~~
    ~~process-identifier [89] Unsigned OPTIONAL~~
    ~~priority [86] Unsigned OPTIONAL~~
    ~~issue-confirmed-notifications [51] BOOLEAN OPTIONAL~~
    event-time-stamps [130] SEQUENCE OF BACnetTimeStamp, -- accessed as a BACnetARRAY
    profile-name [167] CharacterString OPTIONAL
    }

[Change **Annex D.11**, the 2nd example of an Event Enrollment object, p.447]

D.11 Example of an Event Enrollment Object

…
Example 2: CHANGE_OF_VALUE_EVENT event type.

This is an example of a CHANGE_OF_VALUE Event_Type. The management of recipients is done through the use of optional properties of the Event Enrollment object instead of a Notification Class object.

EVENT_ENROLLMENT

Property:	Object_Identifier =	(Event Enrollment, Instance 2)
Property:	Object_Name =	"Zone1TempCOV"
Property:	Object_Type =	EVENT_ENROLLMENT
Property:	Description =	"Zone 1 Temperature COV"
Property:	Event_Type =	CHANGE_OF_VALUE
Property:	Notify_Type =	EVENT
Property:	Event_Parameters =	(5, 0.25)
Property:	Object_Property_Reference =	((Device, Instance 12),(Analog Input, Instance 2), Present_Value)
Property:	Event_State =	NORMAL
Property:	Event_Enable =	(TRUE, FALSE, FALSE)
Property:	Acked_Transitions =	(TRUE, TRUE, TRUE)
Property:	Notification_Class =	3
~~Property:~~	~~Recipient =~~	~~(Device, Instance 33)~~
~~Property:~~	~~Process_Identifier =~~	~~4~~
~~Property:~~	~~Priority =~~	~~87~~
~~Property:~~	~~Issue_Confirmed_Notifications =~~	~~TRUE~~
Property:	Event_Time_Stamps =	((23-MAR-95,18:50:21.2), (*-*-*,*:*:*.*), (23-MAR-95,19:01:34.0))

135*a*-5. Add the capability to issue I-Am responses on behalf of MS/TP slave devices.

Addendum 135*a*-5

[Append to **Clause 12**, para. 2, p. 127]

12. MODELING CONTROL DEVICES AS A COLLECTION OF OBJECTS

…

All objects are referenced by their Object_Identifier property. Each object within a single BACnet Device shall have a unique value for the Object_Identifier property. When combined with the system-wide unique Object_Identifier of the BACnet Device, this provides a mechanism for referencing every object in the control system network. *No object shall have an Object_Identifier with an instance number of 4194303. Object properties that contain BACnetObjectIdentifiers may use 4194303 to indicate that the property is not initialized.*

[Append to **Clause 15.5.2**, p. 286, ReadProperty service]

15.5.2 Service Procedure

After verifying the validity of the request, the responding BACnet-user shall attempt to access the specified property of the specified object. If the access is successful, a `Result(+)' primitive, which returns the accessed value, shall be generated. If the access fails, a `Result(-)' primitive shall be generated.

When the object-type in the Object Identifier parameter contains the value 'Device Object' and the instance in the 'Object Identifier' parameter contains the value 4194303, the responding BACnet-user shall treat the Object Identifier as if it correctly matched the local Device object. This allows the device instance of a device that does not generate I-Am messages to be determined.

[Append to **Clause 15.7.2**, p. 293, ReadPropertyMultiple service]

15.7.2 Service Procedure

… which returns all accessed values and error codes for all properties that could not be accessed.

When the object-type in the Object Identifier portion of the Read Access Specification parameter contains the value 'Device Object' and the instance of that 'Object Identifier' parameter contains the value 4194303, the responding BACnet-user shall treat the Object Identifier as if it correctly matched the local Device object. This allows the device instance of a device that does not generate I-Am messages to be determined.

[Change **Table 12-12**, p. 169]

Table 12-12. Properties of the Device Object Type

Property Identifier	Property Datatype	Conformance Code
…	…	…
Slave_Proxy_Enable	*BACnetArray[N] of BOOLEAN*	*O*[10]
Manual_Slave_Address_Binding	*List of BACnetAddressBinding*	*O*[10]
Auto_Slave_Discovery	*BACnetArray[N] of BOOLEAN*	*O*[11]
Slave_ Address_Binding	*List of BACnetAddressBinding*	*O*[12]
Profile_Name	CharacterString	O

…

[9] This property is required if the device supports execution of either the SubscribeCOV or SubscribeCOVProperty service.

[10] *This property shall be present and writable if the device is capable of being a Slave-Proxy device.*

[11] *This property shall be present if the device is capable of being a Slave-Proxy device that implements automatic discovery of slaves.*

[12] *This property shall be present if the device is capable of being a Slave-Proxy device.*

[Renumber **Clause 12.10.39** to **12.10.43**, p.173]

[Insert new **Clauses 12.10.39 through 12.10.42**, p. 173]

12.10.39 Slave_Proxy_Enable

This property, of type BACnetArray of BOOLEAN, is an indication whether (TRUE) or not (FALSE) the device will perform Slave-Proxy functions for each of the MS/TP ports represented by each array element. This property shall be present if the device is capable of performing the functions of a Slave-Proxy device. The value of this property shall be retained over a device reset.

12.10.40 Manual_Slave_Address_Binding

This property, of type List of BACnetAddressBinding, describes the manually configured set of slave devices for which this device is acting as a Slave Proxy as described in 16.9.2. This property shall be present if the device is capable of performing the functions of a Slave-Proxy device. If present, and the device is directly attached to an MS/TP network, then this property shall be writable.

This property is used to manually configure a set of slave devices for which this device will be a proxy. This property allows a Slave Proxy that does not support automatic slave discovery be configured with a set of slaves for which this device will be a proxy. It also allows a Slave-Proxy device to be a proxy for Slave devices that do not support the special object instance of 4194303 as described in Clause 12. The value of this property shall be retained over a device reset. When enabled, the Slave-Proxy device shall periodically check each device that is in this list, and not in the Slave_Address_Binding list, by reading the device's Protocol_Services_Supported property from the device's Device object using the ReadProperty service. If the device responds and indicates that it does not execute the Who-Is service, it shall be added to the Slave_Address_Binding property. The period at which the devices are checked is a local matter.

12.10.41 Auto_Slave_Discovery

This property, of type BACnetArray of BOOLEAN, is an indication whether (TRUE) or not (FALSE) the device will perform automatic slave detection functions for each of the MS/TP ports represented by each array element. This property shall be present if the device is capable of performing the functions of a Slave-Proxy device. The value of this property shall be retained over a device reset.

Slave detection shall be accomplished by the proxy device using ReadProperty services to read, at a minimum, the Device object's Protocol_Services_Supported property for each MAC address on each port where Auto_Slave_Discovery for that port is TRUE. The ReadProperty service shall use the special object instance of 4194303 as described in Clause 12. If the device is found to support execution of the Who-Is service, it is ignored; otherwise, the device shall be added to the Slave_Address_Binding property. The slave detection algorithm shall be repeated periodically. The period at which it is repeated is a local matter.

12.10.42 Slave_Address_Binding

This property, of type List of BACnetAddressBinding, describes the set of slave devices for which this device is acting as a Slave-Proxy as described in 16.9.2. This property shall be present if the device is capable of performing the functions of a Slave-Proxy device. If present, and the device is directly attached to an MS/TP network, then this property shall be writable.

The set of devices described by the Slave_Address_Binding property consists of those devices described in the Manual_Slave_Address_Binding and those devices that are automatically discovered. When enabled, the Slave-Proxy device shall periodically check each device in this list by reading the device's Protocol_Services_Supported property from the device's Device object using the ReadProperty service. If the device fails to respond, or indicates that it executes Who-Is, it shall be removed from the list. The period at which the devices are checked is a local matter.

[Change **Clause 16.10.2**, p. 318]

16.10.2 Who-Is Service Procedure

The sending BACnet-user shall transmit the Who-Is unconfirmed request, normally using a broadcast address. If the 'Device Instance Range Low Limit' and 'Device Instance Range High Limit' parameters are omitted, then all receiving BACnet-users shall return their Device Object_Identifier in individual responses using the I-Am service. If the 'Device Instance Range Low Limit' and 'Device Instance Range High Limit' parameters are present, then only those receiving BACnet-users whose Device Object_Identifier instance number falls within the range 'Device Instance Range Low Limit ≤ Device Object_Identifier Instance Number ≤ 'Device Instance Range High Limit' shall return their Device Object_Identifier using the I-Am service.

If the receiving BACnet-user has a Slave_Proxy_Enable property and the Slave_Proxy_Enable for the receiving port is TRUE, then the BACnet-user shall respond with an I-Am unconfirmed request for each of the slave devices on the MS/TP network that are present in the Slave_Address_Binding property and that match the device range parameters. The I-Am unconfirmed requests that are generated shall be generated as if the slave device originated the service. If the I-Am confirmed request is to be placed onto the MS/TP network on which the slave resides, then the MAC address included in the packet shall be that of the slave device. In the case where the I-Am confirmed request is to be placed onto a network other than that on which the slave resides, then the network layer shall contain SLEN and SNET filled in with the slave's MAC address as if it were routing a packet originally generated by the slave device.

[Change BACnetPropertyIdentifier production in **Clause 21**, p. 400]

BACnetPropertyIdentifier ::= ENUMERATED {

…
attempted-samples (124),
auto-slave-discovery (169)
average-value (125),
…
manipulated-variable-reference (60),
manual-slave-address-binding (170),
maximum-output (61),
…
setpoint-reference (109),
slave-address-binding (171),
slave-proxy-enable (172),
state-text (110),
…
-- see profile-name (168),
-- see auto-slave-discovery (169),
-- see manual-slave-address-binding (170),
-- see slave-address-binding (171),
-- see slave-proxy-enable (172),
…
}

-- The special property identifiers all, optional, and required are reserved for use in the ReadPropertyConditional and
-- ReadPropertyMultiple services or services not defined in this standard.
--
-- Enumerated values 0-511 are reserved for definition by ASHRAE. Enumerated values 512-4194303 may be used by
-- others subject to the procedures and constraints described in Clause 23. The highest enumeration used in this version is ~~168~~. *172.*

[Change **Annex C**, p. 432]

DEVICE ::= SEQUENCE {

. . .

slave-proxy-enable	*[172]*	*SEQUENCE OF BOOLEAN OPTIONAL,*
auto-slave-discovery	*[169]*	*SEQUENCE OF BOOLEAN OPTIONAL,*
slave-address-binding	*[171]*	*SEQUENCE OF BACnetAddressBinding OPTIONAL,*
manual-slave-address-binding	*[170]*	*SEQUENCE OF BACnetAddressBinding OPTIONAL*,
profile-name	[168]	CharacterString OPTIONAL

}

135*a*-6. Add a new silenced mode to the DeviceCommunicationControl service.

Addendum 135*a*-6

[Change **Clause 16.1**, p. 304]

16.1 DeviceCommunicationControl Service

The DeviceCommunicationControl service is used by a client BACnet-user to instruct a remote device to stop initiating and *optionally stop* responding to all APDUs (except DeviceCommunicationControl or ReinitializeDevice) on the communication network or internetwork for a specified duration of time. This service is primarily used by a human operator for diagnostic purposes. A password may be required from the client BACnet-user prior to executing the service. The time duration may be set to "indefinite," meaning communication must be re-enabled by a DeviceCommunicationControl or ReinitializeDevice service, not by time.

[Change **Clause 16.1.1.1.2**, p. 304]

16.1.1.1.2 Enable/Disable

This parameter is an enumeration that may take on the values ENABLE ~~or~~, DISABLE*, or DISABLE_INITIATION*. It is used to indicate whether the responding BACnet-user is to enable *all, disable initiation* or disable *all* communications on the network interface. When this parameter has a value of ENABLE, communications shall be enabled. When this parameter has a value of DISABLE, *all* communications shall be disabled. *When this parameter has a value of DISABLE_INITIATION, the initiation of communications shall be disabled.* When network communications are *completely* disabled, only DeviceCommunicationControl and ReinitializeDevice APDUs shall be processed and no messages shall be initiated. *When the initiation of communications is disabled, all APDUs shall be processed and responses returned as required and no messages shall be initiated with the exception of I-Am requests, which shall be initiated only in response to Who-Is messages. In this state, a device that supports I-Am request initiation shall send one I-Am request for any Who-Is request that is received if and only if the Who-Is request does not contain an address range or the device is included in the address range.*

[Change DeviceCommunicationControl-Request production in **Clause 21**, p. 379]

```
DeviceCommunicationControl-Request  ::= SEQUENCE {
    timeDuration        [0]  Unsigned16 OPTIONAL,
    enable-disable      [1]  ENUMERATED {
                                 enable    (0),
                                 disable   (1) (1),
                                 disable-initiation (2)
                                 },
    password            [2] CharacterString (SIZE(1..20)) OPTIONAL
    }
```

135*a*-7. Add 21 new engineering units.

Addendum 135*a*-7

[Change BACnetEngineeringUnits production in **Clause 21**, p.389]

```
Units  ::= ENUMERATED {
        …
--Electrical
        …
        ohms                                (4),
        milliohms                           (145),
        kilohms                             (122),
        …
--Energy
        …
        kilowatt-hours                      (19),
        megawatt-hours                      (146),
        btus                                (20),
        kilo-btus                           (147),
        mega-btus                           (148),
        ...
--Enthalpy
        joules-per-kilogram-dry-air         (23),
        kilojoules-per-kilogram-dry-air     (149),
        megajoules-per-kilogram-dry-air     (150),
        btus-per-pound-dry-air              (24),
        …
--Entropy
        joules-per-degree-Kelvin            (127),
        kilojoules-per-degree-Kelvin        (151),
        megajoules-per-degree-Kelvin        (152),
        joules-per-kilogram-degree-Kelvin   (128),

--Force
        newton                              (153),

--Frequency
        cycles-per-hour                     (25),
        …
--Mass Flow
        grams-per-second                    (154),
        grams-per-minute                    (155),
        kilograms-per-second                (42),
        …
        pounds-mass-per-hour                (46),
        tons-per-hour                       (156),

--Power
        …
        btus-per-hour                       (50),
        kilo-btus-per-hour                  (157),
        horsepower                          (51),
        …
```

```
    --Time
            …
            seconds                 (73),
            hundredths-seconds      (158),
            milliseconds            (159),

    --Torque
            newton-meters           (160),

    --Velocity
            millimeters-per-second  (161),
            millimeters-per-minute  (162),
            meters-per-second       (74),
            meters-per-minute       (163),
            meters-per-hour         (164),
            kilometers-per-hour     (75),
            …
    --Volumetric Flow
            …
            cubic-meters-per-second (85),
            cubic-meters-per-minute (165),
            cubic-meters-per-hour   (135),
                …
        }
-- Enumerated values 0-255 are reserved for definition by ASHRAE. Enumerated values
-- 256-65535 may be used by others subject to the procedures and constraints described
-- in Clause 23. The last enumeration used in this version is ~~144~~ 165.
```

135*a*-8. Specify the behavior of a BACnetArray when its size is changed.

Addendum 135*a*-8

[Change **Clause 12**, p.128]

12. MODELING CONTROL DEVICES AS A COLLECTION OF OBJECTS

…

A "BACnetARRAY" datatype is a structured datatype consisting of an ordered sequence of data elements, each having the same datatype. The components of an array property may be individually accessed (read or written) using an "array index," which is an unsigned integer value. An index of 0 (zero) shall specify that the count of the number of data elements be returned. If the array index is omitted, it means that all of the elements of the array are to be accessed. An array index N, greater than zero, shall specify the Nth element in the sequence. When array properties are used in BACnet objects, the notation "BACnetARRAY[N] of datatype" shall mean an ordered sequence of N data elements, each of which has that datatype. *If the size of an array may be changed by writing to the array, then array element 0 shall be writable. If the value of array element 0 is decreased, the array shall be truncated and the elements of the array with an index greater than the new value of array element 0 are deleted. If the value of array element 0 is increased, the new elements of the array, those with an index greater than the old value of array element 0, shall be created; the values that are assigned to those elements shall be a local matter except where otherwise specified. Where the size of an array is allowed to be changed, writing the entire array as a single property with a different number of elements shall cause the array size to be changed. An attempt to write to an array element with an index greater than the size of the array shall result in an error and shall not cause the array to grow to accommodate the element. Arrays whose sizes are fixed by the Standard shall not be resizable.*

A "List of" datatype is a structured datatype consisting of a sequence of zero or more data elements, each having the same datatype. The length of each "List of" may be variable. Unless specified for a particular use, no maximum size should be assumed for any "List of" implementation. The notation "List of datatype" shall mean a sequence of zero or more data elements, each of which has the indicated type.

The difference between a "BACnetARRAY" property and a "List of" property is that the elements of the array can be uniquely accessed by an array index while the elements of the "List of" property cannot. Moreover, the number of elements in the BACnetARRAY may be ascertained by ~~the use of~~ *reading* the array index 0, while the number of elements present in a "List of" property can only be determined by reading the entire property value and performing a count.

…

[Change **Clause 12.9.8**, Command object, p.168]

12.9.8 Action

This property, of type BACnetARRAY of BACnetActionList, specifies an array of "action lists." These action lists are indexed by the value that is written to the Present_Value property. A given list may be empty, in which case no action takes place, except that In_Process is returned to FALSE and All_Writes_Successful is set to TRUE. If the list is not empty, then for each action in the list, the Command object shall write a particular value to a particular property of a particular object in a particular BACnet Device based on the specifications in each BACnetActionCommand. Each write shall occur in the order that BACnetActionCommand list elements would appear if the list was read using the ReadProperty service. The value zero is a special case that takes no action and behaves like an empty list. The number of defined action lists may be found by reading the Action property with an array index of zero. *If the size of this array is changed, the size of the Action_Text array, if present, shall also be changed to the same size.*

…

[Change **Clause 12.9.9**, Command object, p.168]

12.9.9 Action_Text

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

[Change **Clause 12.10.16**, Device object, p.171]

12.10.16 Object_List

This *read-only* property is a BACnetARRAY of Object_Identifiers, one Object_Identifier for each object within the device that is accessible through BACnet services.

[Change **Clause 12.17.11**, Multi-state Input object, p.205]

12.17.11 Number_Of_States

This property, of type Unsigned, defines the number of states that the Present_Value may have. The Number_of_States property shall always have a value greater than zero. *If the value of this property is changed, the size of the State_Text array, if present, shall also be changed to the same value.*

[Change **Clause 12.17.12**, Multi-state Input object, p.205]

12.17.12 State_Text

This property is a BACnetARRAY of character strings representing descriptions of all possible states of the Present_Value. The number of descriptions matches the number of states defined in the Number_of_States property. The Present_Value, interpreted as an integer, serves as an index into the array. *If the size of this array is changed, the Number_Of_States property shall also be changed to the same value.*

[Change **Clause 12.18.11**, Multi-state Output object, p.208]

12.18.11 Number_Of_States

This property, of type Unsigned, defines the number of states that the Present_Value may have. The Number_of_States property shall always have a value greater than zero. *If the value of this property is changed, the size of the State_Text array, if present, shall also be changed to the same value.*

[Change **Clause 12.18.12**, Multi-state Output object, p.209]

12.18.12 State_Text

This property is a BACnetARRAY of character strings representing descriptions of all possible states of the Present_Value. The number of descriptions matches the number of states defined in the Number_of_States property. The Present_Value, interpreted as an integer, serves as an index into the array. *If the size of this array is changed, the Number_Of_States property shall also be changed to the same value.*

[Change **Clause 12.19.10**, Multi-state Value object, p.213]

12.19.10 Number_Of_States

This property, of type Unsigned, defines the number of states that the Present_Value may have. The Number_of_States property shall always have a value greater than zero. *If the value of this property is changed, the size of the State_Text array, if present, shall also be changed to the same value.*

[Change **Clause 12.19.11**, Multi-state Value object, p.213]

12.19.11 State_Text

This property is a BACnetARRAY of character strings representing descriptions of all possible states of the Present_Value. The number of descriptions matches the number of states defined in the Number_of_States property. The Present_Value, interpreted as an integer, serves as an index into the array. *If the size of this array is changed, the Number_Of_States property shall also be changed to the same value.*

[Append to **Clause 12.22.8**, Schedule object, p.225]

12.22.8 Exception_Schedule

…

If a BACnet Device supports writing to the Exception_Schedule property, all possible choices in the BACnetSpecialEvents shall be supported. *If the size of this array is increased by writing to array index zero, each new array element shall contain an empty List of BACnetTimeValue.*

135*a*-9. Clarify the behavior of a BACnet router when it receives an unknown network message type.

Addendum 135*a*-9

[Change **Clause 6.6.3.5**, p.61]

6.6.3.5 Reject-Message-To-Network

Reject-Message-To-Network is generated by a router when it receives a message that it is unable to relay to the DNET specified in the NPCI, it receives an unknown network layer message *directed specifically to that router*, or a directly connected PTP link is disconnected. The reasons for rejecting the message are set forth in 6.4.4.
…

[Add a new entry to **History of Revisions**, p.557]

(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	4	**Addendum *a* to ANSI/ASHRAE 135-2001** Approved by the ASHRAE Standards Committee January XX, 2004 and by the ASHRAE Board of Directors January XX, 2004; and by the American National Standards Institute XXX, 2004. 1. Add Partial Day Scheduling to the Schedule object. 2. Enable reporting of proprietary events by the Event Enrollment object. 3. Allow detailed error reporting when all ReadPropertyMultiple accesses fail. 4. Remove the Recipient property from the Event Enrollment object. 5. Add the capability to issue I-Am responses on behalf of MS/TP slave devices. 6. Add a new silenced mode to the DeviceCommunicationControl service. 7. Add 21 new engineering units. 8. Specify the behavior of a BACnetArray when its size is changed. 9. Clarify the behavior of a BACnet router when it receives an unknown network message type.
1	4	**Addendum *c* to ANSI/ASHRAE 135-2001** Approved by the ASHRAE Standards Committee January XX, 2004 and by the ASHRAE Board of Directors January XX, 2004; and by the American National Standards Institute XXX, 2004. 1. Allow Life Safety objects to advertise supported mode. 2. Add Unsilence Options to the LifeSafetyOperation Service. 3. Specify the relationship between the Event_Type and Event_Parameter properties. 4. Add a new Accumulator Object Type. 5. Add a new Pulse Converter Object Type. 6. Standardize event notification priorities. 7. Define Abort reason when insufficient segments are available. 8. Add new Error Codes and specify usage.

1	4	**Addendum *d* to ANSI/ASHRAE 135-2001** Approved by the ASHRAE Standards Committee January XX, 2004 and by the ASHRAE Board of Directors January XX, 2004; and by the American National Standards Institute XXX, 2004. 1. Add clauses describing BACnet-EIB/KNX mapping.

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.