



# ASHRAE ADDENDA

## BACnet<sup>®</sup> —A Data Communication Protocol for Building Automation and Control Networks

Approved by the ASHRAE Standards Committee on January 29, 2011; by the ASHRAE Board of Directors on February 2, 2011; and by the American National Standards Institute on February 3, 2011.

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](http://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](http://www.ashrae.org)) or from ASHRAE Customer Service, 1791 Tullie Circle, NE, Atlanta, GA 30329-2305. E-mail: [orders@ashrae.org](mailto: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](http://www.ashrae.org/permissions).

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.



ISSN 1041-2336

**American Society of Heating, Refrigerating  
and Air-Conditioning Engineers, Inc.**  
1791 Tullie Circle NE, Atlanta, GA 30329  
[www.ashrae.org](http://www.ashrae.org)

**ASHRAE Standing Standard Project Committee 135**  
**Cognizant TC: TC 1.4, Control Theory and Application**  
**SPLS Liaison: Richard L. Hall**

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

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

---

**ASHRAE STANDARDS COMMITTEE 2010–2011**

H. Michael Newman, <i>Chair</i>	Allan B. Fraser	Janice C. Peterson
Carol E. Marriott, <i>Vice-Chair</i>	Krishnan Gowri	Douglas T. Reindl
Douglass S. Abramson	Maureen Grasso	Boggarm S. Setty
Karim Amrane	Cecily M. Grzywacz	James R. Tauby
Robert G. Baker	Richard L. Hall	James K. Vallort
Hoy R. Bohanon, Jr.	Nadar R. Jayaraman	William F. Walter
Steven F. Bruning	Byron W. Jones	Michael W. Woodford
Kenneth W. Cooper	Jay A. Kohler	Craig P. Wray
Martin Dieryckx	Frank Myers	Hugh F. Crowther, <i>BOD ExO</i>
		William P. Bahnfleth, <i>CO</i>

Stephanie 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 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, or
- d. permission to reprint portions of the Standard.

**DISCLAIMER**

ASHRAE uses its best efforts to promulgate Standards and Guidelines for the benefit of the public in light of available information and accepted industry practices. However, ASHRAE does not guarantee, certify, or assure the safety or performance of any products, components, or systems tested, installed, or operated in accordance with ASHRAE's Standards or Guidelines or that any tests conducted under its Standards or Guidelines will be nonhazardous or free from risk.

**ASHRAE INDUSTRIAL ADVERTISING POLICY ON STANDARDS**

ASHRAE Standards and Guidelines are established to assist industry and the public by offering a uniform method of testing for rating purposes, by suggesting safe practices in designing and installing equipment, by providing proper definitions of this equipment, and by providing other information that may serve to guide the industry. The creation of ASHRAE Standards and Guidelines is determined by the need for them, and conformance to them is completely voluntary.

In referring to this Standard or Guideline and in marking of equipment and in advertising, no claim shall be made, either stated or implied, that the product has been approved by ASHRAE.

**[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*ac* 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*ac*-1 Clarify the Usage of Dates and Times, 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 ~~strike through~~. Where entirely new subclauses are added, plain type is used throughout.

## 135-2008ac-1 Clarify the Usage of Dates and Times

### Rationale

Dates and Times are used in various ways throughout the standard. In some cases, the use of 'unspecified' or 'wildcard' values is specifically allowed, but in other places, it is ambiguous. This addendum specifies the usage of dates and times more precisely.

[Add new **Clauses 3.2.x** pp. 1-5]

[Note that additional special date values, their description and their encodings were introduced with addendum 135-2008h-10]

**3.2.x date pattern:** *a date that may contain one or more unspecified octets or special date values.*

**3.2.x special date value:** *a date value that is one of the special values such as "even months", "last day of month", etc. These special date values are used in subcomponents (octets) of a value of type Date.*

**3.2.x specific date:** *a fully specified date. For example, January 24, 1991, Day of week = Thursday. A specific date shall contain no unspecified octets or Special Date Values.*

**3.2.x specific datetime:** *a BACnetDateTime construct composed of a specific date and a specific time.*

**3.2.x specific time:** *a fully specified time. For example, 17:35:45.17 (= 5:35:45.17 P.M.). A specific time shall contain no unspecified octets.*

**3.2.x time pattern:** *a time that may contain one or more unspecified octets.*

**3.2.x unspecified date:** *a date composed entirely of unspecified octets (A value of X'FF' = D'255').*

**3.2.x unspecified datetime:** *a BACnetDateTime construct composed of an unspecified date and an unspecified time.*

**3.2.x unspecified octet:** *an octet used in the context of date, time or BACnetWeekNDay values that contains the value X'FF' = D'255'.*

**3.2.x unspecified time:** *a time composed entirely of unspecified octets (A value of X'FF' = D'255').*

[Change **Clause 3.2.57**. p. 5]

**3.2.57 timestamp:** *the indication of the point in time ~~date and time~~ recorded for and accompanying the record of an event or operation.*

[Change **Clause 12** pp. 128-129]

...

Several object types defined in this clause— the Command, Event Enrollment, Group, Loop, and Schedule— have one or more properties of type BACnetObjectPropertyReference. The property identifier component of these references may not be the special property identifiers ALL, REQUIRED, or OPTIONAL. These are reserved for use in the ReadPropertyConditional and ReadPropertyMultiple services or in services not defined in this standard.

*Several object types defined in this clause have properties that are of type BACnetDateTime, and specify a specific point in time. These properties shall have an unspecified datetime value if the point in time is undefined or a specific datetime value if the point in time is specified.*

*There are a number of objects defined in this clause that have properties of the type BACnetDateRange, for example the Date\_List property in the Calendar object, whose construct includes a startDate and an endDate. Both startDate and endDate may be an unspecified date or a specific date only. For purposes of comparing date ranges, the following logic shall be applied.*

*The use of an unspecified date in the startDate means "any date up to and including the endDate." The use of an unspecified date in the endDate means "any date after and including the startDate." The use of an unspecified date in both the startDate and the endDate means "any date" or "always."*

*Several object types defined in this clause have properties that contain timestamp values. If no event or operation has yet occurred, then timestamp values of type BACnetDateTime shall have an unspecified datetime value, timestamp values of type Time shall have an unspecified time value, and timestamp values of type Unsigned shall have a value of zero. If the event or operation has occurred, then the timestamp value shall have a specific datetime value, a specific time value, or a value greater than zero, respectively. If a device supports the Local\_Date and Local\_Time properties, then all timestamps created by the device shall use the BACnetDateTime form.*

Several object types defined in this clause have a property called "Reliability." This property is an enumerated datatype that may have different possible enumerations for different object types. The values defined below are a superset of all possible values of the Reliability property for all object types. The range of possible values returned for each specific object is defined in the appropriate object type definition.

...

[Change all **Clauses 12.X.Y Event\_Time\_Stamps**, in the standard and all addenda by removing the sentence shown]  
[Note that **Clause 12.12.13**, which has 'FF' instead of X'FF', has been corrected with by a pending erratum.]

### **12.X.Y Event\_Time\_Stamps**

~~... Time stamps of type Time or Date shall have X'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. ...~~

[Change **Clause 12.1.15**, p.136]

### **12.1.15 Value\_Change\_Time**

This read-only property, of type BACnetDateTime, shall be present if the Present\_Value property is adjustable by writing to the Value\_Before\_Change or Value\_Set properties. It represents the date and time of the most recent occurrence of such a write operation. ~~If no such write has yet occurred, this property shall have wildcard values for all date and time fields.~~ *This property shall have an unspecified datetime to indicate that it is uninitialized; otherwise, it shall have a specific datetime value.*

[Change **Clause 12.1.18**, p. 136]

### **12.1.18 Logging\_Record**

This read-only property, of type BACnetAccumulatorRecord, is a list of values that must be acquired and returned "atomically" in order to allow proper interpretation of the data.

If the Logging\_Object property is present, then, when Logging\_Record is acquired by the object identified by Logging\_Object, this list of values shall be saved and returned when read by other objects or devices. If the Logging\_Object property is present and Logging\_Record has not yet been acquired by the object identified by Logging\_Object, 'timestamp' shall contain ~~all wildcards~~, *an unspecified datetime*, 'present-value' and 'accumulated-value' shall contain the value zero, and 'accumulator-status' shall indicate STARTING.

...

**STARTING** This value indicates that the data in Logging\_Records is either the first data to be acquired since startup by the object identified by Logging\_Object (if 'timestamp' has ~~non-wildcard values a specific datetime~~) or that no data has been acquired since startup by the object identified by Logging\_Object (in which case 'timestamp' has ~~all wildcard values an unspecified datetime~~).

...

[Change **Clause 12.9.6**, p.174]

**12.9.6 Date\_List**

This property is a List of BACnetCalendarEntry, each of which is either ~~an individual date a specific date or date pattern~~ (Date), range of dates (BACnetDateRange), or month/week-of-month/day-of-week specification (BACnetWeekNDay). If the current date matches the calendar entry criteria, the present value of the Calendar object is TRUE. ~~Individual fields of the various constructs may also be unspecified in which case the field acts as a "wildcard" for determining if the current date results in a match. In a date range, for example, if the startDate is unspecified, it means "any date up to and including the endDate." If the endDate is unspecified, it means "any date from the startDate on."~~

*As an example, if ~~the calendar entry is were~~ a BACnetWeekNDay with ~~unspecified an unspecified octet in the month and week-of-month fields but with a specific day-of-week, it means that would mean~~ the Calendar object is ~~would be~~ TRUE on that day-of-week all year long. ~~If a BACnet Device permits writing to the Date\_List property, all choices in the BACnetCalendarEntry shall be permitted.~~*

*If a BACnet Device permits writing to the Date\_List property, all choices and all allowed forms of values in the BACnetCalendarEntry shall be supported.*

[Change **Table 12-13**, pp 180-181]

Property Identifier	Property Datatype	Conformance Code
...		
Local_Time	Time	O <sup>3,4,15</sup>
Local_Date	Date	O <sup>3,4,15</sup>
...		

<sup>15</sup> *These properties shall be present if the device is capable of tracking date and time.*

[Change **Clauses 12.11.22 and 12.11.23**, p.183]

**12.11.22 Local\_Time**

The Local\_Time property, of type Time, shall indicate the time of day to the best of the device's knowledge. If the BACnet Device does not have any knowledge of time or date, then the Local\_Time property shall be omitted. *This property shall be present if the BACnet Device is capable of tracking date and time. If the device restarts, and there is no local time available yet, then the clock shall be initialized to the time 00:00:00.00 and the device shall commence tracking the time.*

**12.11.23 Local\_Date**

The Local\_Date property, of type Date, shall indicate the date to the best of the device's knowledge. If the BACnet Device does not have any knowledge of time or date, then the Local\_Date property shall be omitted. *This property shall be present if the BACnet Device is capable of tracking date and time. If the device restarts, and there is no local date*

*available yet, then the clock shall be initialized to a date on or before January 1, 1990 and the device shall commence tracking the date.*

[Change **Clause 12.13.7**, p 194]

**12.13.7 Modification\_Date**

This property, of type BACnetDateTime, indicates the last time this object was modified. A File object shall be considered modified when it is created or written to. *If the Local\_Time and Local\_Date properties are present, this property shall contain a specific datetime value; otherwise this property shall contain either a specific or an unspecified datetime value.*

[Change **Clause 12.21**, p. 229]

**12.21 Notification Class Object Type**

...

It is often necessary for event notifications to be sent to multiple destinations or to different destinations based on the time of day or day of week. Notification Classes may specify a list of destinations, each of which is qualified by time, day of week, and type of handling. A destination specifies a set of days of the week (Monday through Sunday) during which the destination is considered viable by the Notification Class object. In addition, each destination has a FromTime and ToTime, which specify a window *using specific times*, on those days of the week, during which the destination is viable. If an event that uses a Notification Class object occurs and the day is one of the days of the week that is valid for a given destination and the time is within the window specified in the destination, then the destination shall be sent a notification. Destinations may be further qualified, as applicable, by any combination of the three event transitions TO-OFFNORMAL, TO-FAULT, or TO-NORMAL.

...

[Change **Clause 12.21.8**, p. 230]

**12.21.8 Recipient\_List**

This property, of type List of BACnetDestination, shall convey a list of one or more recipient destinations to which notifications shall be sent when event-initiating objects using this class detect the occurrence of an event. The destinations themselves define a structure of parameters that is summarized in Table 12-25.

**Table 12-25.** Components of a BACnetDestination

Parameter	Type	Description
Valid Days	BACnetDaysOfWeek	The set of days of the week on which this destination <u>may</u> be used between From Time and To Time
From Time, To Time	Time	The window of time (inclusive) during which the destination is viable on the days of the week <i>specified by Valid Days. These values shall be specific times.</i>
Recipient	BACnetRecipient	The destination device(s) to receive notifications
Process Identifier	Unsigned32	The handle of a process within the recipient device that is to receive the event notification
Issue Confirmed Notifications	Boolean	(TRUE) if confirmed notifications are to be sent and (FALSE) if unconfirmed notifications are to be sent
Transitions	BACnet Event Transition Bits	A set of three flags that indicate those transitions {TO-OFFNORMAL, TO-FAULT, TO-NORMAL} for which this recipient is suitable

[Change **Clauses 12.23.15 and 12.23.16**, p. 241]

### **12.23.15 Update\_Time**

This read-only property, of type BACnetDateTime, reflects the date and time of the most recent change to the Count property as a result of input pulse accumulation and is updated atomically with the Count property. ~~If no such change has yet occurred, this property shall have wildcard values for all date and time fields.~~

### **12.23.16 Count\_Change\_Time**

This read-only property, of type BACnetDateTime, represents the date and time of the most recent occurrence of a write to the Adjust\_Value property. ~~If no such write has yet occurred, this property shall have wildcard values for all date and time fields.~~

[Change **Clause 12.24.7**, p. 245]

### **12.24.7 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 time portion of each BACnetTimeValue shall contain a specific time.* 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 12.24.8**, p. 246]

### **12.24.8 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. 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~~ *a specific date or date pattern* (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 if 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." As an example, if~~ ~~If the calendar entry is were a BACnetWeekNDay with wildcard an~~

*unspecified octet* for month and week-of-month fields but with a specific day-of-week, it ~~means would mean~~ the ~~exception schedule~~ ~~Exception\_Schedule~~ ~~applies would apply~~ on that day-of-week all year long.

*Each element of the list of BACnetTimeValues specifies a time and a value of primitive datatype that is used at the time specified by the time portion. The time portion shall contain a specific time.*

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 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 EventPriority values. If multiple overlapping SpecialEvents have the same EventPriority value, then the SpecialEvent with the lowest index number in the array shall 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 size of this array is increased by writing to array index zero, each new array element shall contain an empty List of BACnetTimeValue.

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.

[Change **Clauses 12.25.5, 12.25.6 and 12.25.7**, p 251]

#### **12.25.5 Enable**

This property, of type BOOLEAN, indicates and controls whether (TRUE) or not (FALSE) logging of events and collected data is enabled. Logging occurs if and only if Enable is TRUE, Local\_Time is on or after Start\_Time, and Local\_Time is before Stop\_Time. If Start\_Time contains ~~any wildcards~~, *an unspecified datetime*, then it shall be considered equal to 'the start of time'. If Stop\_Time contains ~~any wildcards~~, *an unspecified datetime*, then it shall be considered equal to 'the end of time'. Log\_Buffer records of type log-status are recorded without regard to the value of the Enable property.

If Enable is writable, attempts to write the value TRUE to the Enable property while Stop\_When\_Full is TRUE and Record\_Count is equal to Buffer\_Size shall cause a Result(-) response to be issued, specifying an 'Error Class' of OBJECT and an 'Error Code' of LOG\_BUFFER\_FULL.

#### **12.25.6 Start\_Time**

This optional property, of type BACnetDateTime, specifies the date and time at or after which logging shall be enabled by this property. *If this property contains an unspecified datetime, ~~if any of the fields of the BACnetDateTime contain "wildcard" value~~*, then the conditions for logging to be enabled by Start\_Time shall be ignored. If Start\_Time specifies a date and time after Stop\_Time, then logging shall be disabled. This property shall be writable if present.

When Start\_Time is reached, the value of the Enable property is not changed.

### 12.25.7 Stop\_Time

This optional property, of type BACnetDateTime, specifies the date and time at or after which logging shall be disabled by this property. *If this property contains an unspecified datetime* ~~If any of the fields of the BACnetDateTime contain "wildcard" values,~~ then the conditions for logging to be disabled by Stop\_Time shall be ignored. If Stop\_Time specifies a date and time earlier than Start\_Time, then logging shall be disabled. This property shall be writable if present.

When Stop\_Time is reached, the value of the Enable property is not changed.

[Change Clauses 12.27.9 and 12.27.10, p. 266]

### 12.27.8 Enable

This property, of type BOOLEAN, indicates and controls whether (TRUE) or not (FALSE) logging of events is enabled. Logging occurs if and only if Enable is TRUE, Local\_Time is on or after Start\_Time, and Local\_Time is before Stop\_Time. If Start\_Time contains ~~any wildcards,~~ *an unspecified datetime*, then it shall be considered equal to 'the start of time'. If Stop\_Time contains ~~any wildcards,~~ *an unspecified datetime*, then it shall be considered equal to 'the end of time'. Log\_Buffer records of type log-status are recorded without regard to the value of the Enable property.

Attempts to write the value TRUE to the Enable property while Stop\_When\_Full is TRUE and Record\_Count is equal to Buffer\_Size shall cause a Result(-) response to be issued, specifying an 'Error Class' of OBJECT and an 'Error Code' of LOG\_BUFFER\_FULL.

### 12.27.9 Start\_Time

This optional property, of type BACnetDateTime, specifies the date and time at or after which logging shall be enabled by this property. *If this property contains an unspecified datetime,* ~~If any of the fields of the BACnetDateTime contain "wildcard" values,~~ then the conditions for logging to be enabled by Start\_Time shall be ignored. If Start\_Time specifies a date and time after Stop\_Time, then logging shall be disabled. If either of the optional properties Start\_Time or Stop\_Time is present, then both of these properties shall be present. This property shall be writable if present.

### 12.27.10 Stop\_Time

This optional property, of type BACnetDateTime, specifies the date and time at or after which logging shall be disabled by this property. *If this property contains an unspecified datetime,* ~~If any of the fields of the BACnetDateTime contain "wildcard" values,~~ then the conditions for logging to be disabled by Stop\_Time shall be ignored. If Stop\_Time specifies a date and time earlier than Start\_Time, then logging shall be disabled. If either of the optional properties Start\_Time or Stop\_Time is present, then both of these properties shall be present. This property shall be writable if present.

[Change Clause 12.28, p. 272]

## 12.28 Load Control Object Type

...

### CancelShed

If the current time is after Start\_Time plus Shed\_Duration, this request is for an invalid time and is ignored. The object shall stop shedding and enter the SHED\_INACTIVE state.

If Requested\_Shed\_Level is equal to the default value for the choice, or Start\_Time contains ~~wildcards,~~ *an unspecified datetime*, then this is a cancellation of shedding. The object shall stop shedding and enter the SHED\_INACTIVE state.

...

#### FinishedSuccessfulShed

If the current time is after Start\_Time plus Shed\_Duration, the shed request has completed successfully. The object shall stop shedding, set Start\_Time to ~~all-wildcards~~, *an unspecified datetime*, and enter the SHED\_INACTIVE state.

...

[Change **Clause 12.28.11**, p. 275]

#### **12.28.11 Start\_Time**

This property, of type BACnetDateTime, indicates the start of the duty window in which the load controlled by the Load Control object must be compliant with the requested shed. Load shedding (or determination of loads to shed) may need to begin before Start\_Time in order to be compliant with the shed request by Start\_Time. If no shed request is pending or active, Start\_Time shall contain *an unspecified datetime value*. ~~all-wildcard values~~. If a load control command has been issued, and execution of the command has completed, Start\_Time shall be reset by the device to contain *an unspecified datetime value*. ~~all-wildcard values~~. If a client wishes to initiate an immediate shed, it can set Start\_Time to a value prior to the device's current time.

[Change **Clauses 12.30.8, 12.30.9 and 12.30.10**, p. 283]

#### **12.30.8 Enable**

This property, of type BOOLEAN, indicates and controls whether (TRUE) or not (FALSE) logging of events and collected data is enabled. Logging occurs if and only if Enable is TRUE, Local\_Time is on or after Start\_Time, and Local\_Time is before Stop\_Time. If Start\_Time contains ~~any-wildcards~~, *an unspecified datetime*, then it shall be considered equal to 'the start of time'. If Stop\_Time contains ~~any-wildcards~~, *an unspecified datetime*, then it shall be considered equal to 'the end of time'. Log records of type log-status or no-data are recorded without regard to the value of the Enable property.

Attempts to write the value TRUE to the Enable property while Stop\_When\_Full is TRUE and Record\_Count is equal to Buffer\_Size shall cause a Result(-) response to be issued, specifying an 'Error Class' of OBJECT and an 'Error Code' of LOG\_BUFFER\_FULL.

#### **12.30.9 Start\_Time**

This optional property, of type BACnetDateTime, specifies the date and time at or after which logging shall be enabled by this property. *If this property contains an unspecified datetime, if any of the fields of the BACnetDateTime contain* ~~"wildcard" values~~, then the conditions for logging to be enabled by Start\_Time shall be ignored. If Start\_Time specifies a date and time after Stop\_Time, then logging shall be disabled. This property shall be writable if present.

When Start\_Time is reached, the value of the Enable property is not changed.

#### **12.30.10 Stop\_Time**

This optional property, of type BACnetDateTime, specifies the date and time at or after which logging shall be disabled by this property. *If this property contains an unspecified datetime, if any of the fields of the BACnetDateTime contain* ~~"wildcard" values~~, then the conditions for logging to be disabled by Stop\_Time shall be ignored. If Stop\_Time specifies a date and time earlier than Start\_Time, then logging shall be disabled. This property shall be writable if present.

When Stop\_Time is reached, the value of the Enable property is not changed.

[Change **Clause 15.6.3.1.2.3**, p. 347]

**15.6.3.1.2.3 Relation Specifier**

This parameter shall represent one of the six Boolean operators:

- = (equality) > (greater than)
- ≠ (inequality) ≤ (less than or equal to)
- < (less than) ≥ (greater than or equal to)

These Boolean operators may not be used for every property of every object. Some property values may only allow equals (=) and not equals (≠) comparisons. Table 15-8 shows which operators may be used with each datatype.

**Table 15-8. Valid Boolean Operators for BACnet Datatypes**

Datatype	Operators Allowed	Datatype	Operators Allowed
NULL	= ≠	CharacterString	= ≠ < > ≤ ≥
BOOLEAN	= ≠	OCTET STRING	= ≠
Unsigned	= ≠ < > ≤ ≥	ENUMERATED	= ≠
INTEGER	= ≠ < > ≤ ≥	Date <i>Date</i> <sup>1</sup>	= ≠ < > ≤ ≥
REAL	= ≠ < > ≤ ≥	Time	= ≠ < > ≤ ≥
BIT STRING	= ≠	Others	= ≠
BACnetObjectIdentifier	= ≠ < > ≤ ≥		

<sup>1</sup> If a value of type Date contains special date values, comparison for other than equality or inequality shall never match.

[Change **Clause 15.6.3.1.2.4**, p. 347]

...

When properties with a Date or Time datatype are compared for inequalities, the comparison shall be chronological. For example:

$$1\text{-Jan-1991} < 2\text{-Feb-1992}$$

$$7:00:00.00 < 23:00:00.00$$

Subcomponents of a Date or Time, such as the year, may be *an unspecified octet*. ~~"wildcarded" if the value of the subcomponent is the special value "unspecified."~~ Subcomponents of a Date or Time that *are an unspecified octet* have the value "unspecified" shall be skipped in the comparison and that subcomponent shall be considered as "equal." It is possible that either the comparison value, or the property value with which it is being compared, or both, may have subcomponent(s) whose value(s) are *an unspecified octet*. ~~"unspecified."~~ In either case, the rules for matching unspecified *octets* ~~values~~ shall apply. For example (using \* to mean unspecified *octet*):

<u>Comparison Value</u>		<u>Property Value</u>	<u>Because</u>
*-* Sun	=	29-Jan-1995 Sun	Sun = Sun
1-Jan-*	<	2-Feb-1995	1-Jan < 2-Feb
*-Aug-*	>	10-Mar-1995	Aug > Mar
*:59:00.00	>	6:23:99.0	59:00.00 > 23:00.00
6:*.*. *	=	6:25:00.00	6 = 6

*If a Date value has subcomponent(s) whose value(s) are special date values, then comparison for other than equality or inequality shall never match. Comparison with special date values shall result in equality if the respective subcomponent of Date matches the special date or is the same special date value. For the meaning of the special date values, see Clause 20.2.12.*

...

[Change **Clause 15.8.1.1.4.3.1**, p 354]

#### **15.8.1.1.4.3.1 Reference Time**

If 'Count' is positive, the first record to be read shall be the first record with a timestamp newer than the time specified by the 'Reference Time' parameter. If 'Count' is negative, the last record to be read shall be the newest record with a timestamp older than the time specified by the 'Reference Time' parameter. *This parameter shall contain a specific datetime value.*

[Change **Clause 16.7.1.1.1**, p 372]

#### **16.7.1.1.1 Time**

This parameter, of type BACnetDateTime, shall convey the current date and time as determined by the clock in the device issuing the service request. *This parameter shall contain a specific datetime value.*

[Change **Clause 16.8.1.1.1**, p. 373]

#### **16.8.1.1.1 Time**

This parameter, of type BACnetDateTime, shall convey the UTC date and time. *This parameter shall contain a specific datetime value.*

[Change **Clauses 20.2.12 and 20.2.13**, p. 422]

#### **20.2.12 Encoding of a Date Value**

The encoding of a date value shall be primitive, with four contents octets.

Date values shall be encoded in the contents octets as four binary integers. The first contents octet shall represent the year minus 1900; the second octet shall represent the month, with January = 1; the third octet shall represent the day of the month; and the fourth octet shall represent the day of the week, with Monday = 1. A value of X'FF' = D'255' in any of the four octets shall indicate that the corresponding value is ~~unspecified~~ *unspecified and shall be considered a wildcard when matching dates*. If all four octets = X'FF', the corresponding date may be interpreted as "any" or "don't care."

*Neither an unspecified date nor a date pattern shall be used in date values that convey actual dates, such as in a TimeSynchronization-Request.*

*The processing of a day of week received in a service that is in the range 1 to 7 and is inconsistent with the values in the other octets shall be a local matter.*

Example: Application-tagged *specific* date value

ASN.1 =           Date  
Value =           January 24, 1991 (Day of week = Thursday)  
Application Tag = Date (Tag Number = 10)  
Encoded Tag =    X'A4'  
Encoded Data =   X'5B011804'

*Example: Application-tagged date pattern value*

ASN.1 = Date  
 Value = year = 1991, month is unspecified, day = 24, day of week is unspecified  
 Application Tag = Date (Tag Number = 10)  
 Encoded Tag = X'A4'  
 Encoded Data = X'5BFF18FF'

**20.2.13 Encoding of a Time Value**

The encoding of a time value shall be primitive, with four contents octets.

Time values shall be encoded in the contents octets as four binary integers. The first contents octet shall represent the hour, in the 24-hour system (1 P.M. = D'13'); the second octet shall represent the minute of the hour; the third octet shall represent the second of the minute; and the fourth octet shall represent the fractional part of the second in hundredths of a second. A value of X'FF' = D'255' in any of the four octets shall indicate that the corresponding value is ~~unspecified~~. *unspecified and shall be considered a wildcard when matching times.* If all four octets = X'FF', the corresponding time may be interpreted as "any" or "don't care."

*Neither an unspecified time nor a time pattern shall be used in time values that convey actual time, such as those presented by the Local\_Time property of the Device object or in a TimeSynchronization-Request.*

*Example: Application-tagged specific time value*

ASN.1 = Time  
 Value = ~~5:35:45.17 P.M.~~ = 17:35:45.17 (= 5:35:45.17 P.M.)  
 Application Tag = Time (Tag Number = 11)  
 Encoded Tag = X'B4'  
 Encoded Data = X'11232D11'

[Change **Clause 21, BACnetDateRange** production, p.449]

**BACnetDateRange** ::= SEQUENCE { -- see Clause 20.2.12 for restrictions  
     startDate Date,  
     endDate Date  
 }

[Change **Clause 21, BACnetDateTime** production, p.450]

**BACnetDateTime** ::= SEQUENCE {  
     date Date, --see Clause 20.2.12 for restrictions  
     time Time --see Clause 20.2.13 for restrictions  
 }

[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**

<i>Protocol</i>		<i>Summary of Changes to the Standard</i>
<i>Version</i>	<i>Revision</i>	
...	...	...
1	12	<p><b>Addendum ac to ANSI/ASHRAE 135-2008</b>                      Approved by the ASHRAE Standards Committee January 29, 2011; by the ASHRAE Board of Directors February 2, 2011; and by the American National Standards Institute February 3, 2011.</p> <p>1. Clarify the Usage of Dates and Times.</p>

**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.

