



ASHRAE ADDENDA

Method of Test for Conformance to BACnet[®]

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) or in paper form from the Manager of Standards.

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

© 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

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>	David G. Holmberg	David G. Shike
Carl Neilson, <i>Vice-Chair</i>	Robert L. Johnson	Ted Sunderland
Bernhard Isler, <i>Secretary*</i>	Stephen Karg*	William O. Swan, III
Donald P. Alexander*	Simon Lemaire	David B. Thompson*
Barry B. Bridges*	J. Damian Ljungquist*	Daniel A. Traill
Coleman L. Brumley, Jr.	James G. Luth	Stephen J. Treado*
Ernest C. Bryant	John J. Lynch	Klaus Wagner
A. J. Capowski	Brian Meyers	J. Michael Whitcomb*
Clifford H. Copass	Dana Petersen	Grant N. Wichenko*
Sharon E. Dinges*	Carl J. Ruther	Christoph Zeller
Daniel P. Giorgis	Frank Schubert	Scott Ziegenfus

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

FOREWORD

Addendum 135.1*h* to ANSI/ASHRAE Standard 135.1-2009 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.

135.1-2009*h*-1. Change GetEventInformation Chaining Tests, p. 2.

135.1-2009*h*-2. Change CHANGE_OF_STATE Test for Event Enrollment Object, p. 3.

135.1-2009*h*-3. Change ConfirmedCOVEEventNotification Service Initiation Tests to Non-infinite Lifetimes, p. 6.

135.1-2009*h*-4. Change intrinsic tests for Event Enrollment Object, p. 9

In the following document, language added to existing clauses of ANSI/ASHRAE 135.1-2009 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.

135.1-2009h-1 Change GetEventInformation Chaining Tests.

Rationale

The test requires that it be executed using a 50 octet APDU, but the response for a single event can exceed this amount when the timestamps form of the BACnetDateTime are used.

[Change **Clause 9.8.3**, p. 211]

9.8.3 Event Information with Multiple Active Events

...

Configuration Requirements: The IUT shall be configured so that there are more than one active event states, but fewer than would require transmission with 'More Events' = TRUE. *If the IUT cannot be configured to contain multiple active events which can be transmitted in a single GetEventInformation service response without 'More Events' = TRUE, then this test shall be skipped.*

[Change **Clause 9.8.6**, p. 212]

9.8.6 Chaining Test

...

Configuration Requirements: The IUT shall be configured so that there are more event states than can be conveyed in a single APDU of ~~50~~ 128 bytes. *The IUT shall be configured to contain enough events to trigger the chaining effect. If the IUT cannot be configured to contain enough active events to trigger chaining, then this test shall be skipped.*

135.1-2009h-2 Change CHANGE_OF_STATE Test for Event Enrollment Objects.

Rationale

The CHANGE_OF_STATE test was incorrect when used to test an Event Enrollment Object.

[Change **Clause 8.4.2**, p. 119]

[Note to reviewer: When the numbering of the test steps has been changed due to added or deleted steps, the final correct numbering of the steps is shown but the changes to the test-step numbers are not indicated with both strikethrough and italics in order to maintain proper indentation. Revised numbering in steps is indicated by italics only.]

8.4.2 CHANGE_OF_STATE Tests

...

Test Steps:

1. VERIFY Event_State = NORMAL
2. *IF (the object, or referenced object, if using Event Enrollment, is a binary object or it is a multi-state object with a non-empty Alarm_Values property) THEN*

[Indent all statements starting here and through newly numbered line 30. inclusive]

3. IF (Present_Value is writable) THEN
 WRITE Present_Value = (a value x: x = Alarm_Value for binary objects or one of the Alarm_Values for multi-state objects)
ELSE
 MAKE (Present_Value have a value x: x = Alarm_Value for binary objects or one of the Alarm_Values for multi-state objects)
4. WAIT (Time_Delay)
5. BEFORE **Notification Fail Time**
 RECEIVE ConfirmedEventNotification-Request,
 'Process Identifier' = (any valid process ID),
 'Initiating Device Identifier' = IUT,
 'Event Object Identifier' = (the intrinsic reporting object being tested or the Event Enrollment object being tested),
 'Time Stamp' = (the current local time),
 'Notification Class' = (the configured notification class),
 'Priority' = (the value configured to correspond to a TO-OFFNORMAL transition),
 'Event Type' = CHANGE_OF_STATE,
 'Notify Type' = EVENT | ALARM,
 'AckRequired' = TRUE | FALSE,
 'From State' = NORMAL,
 'To State' = OFFNORMAL,
 'Event Values' = Present_Value, Status_Flags
6. TRANSMIT BACnet-SimpleACK-PDU
7. *IF (the object being tested is NOT an Event Enrollment object) THEN*
 VERIFY Status_Flags = (TRUE, FALSE, ?, ?)
8. VERIFY Event_State = OFFNORMAL
9. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
 VERIFY Event_Time_Stamps = (the timestamp in step 45, *, *)
10. IF (Present_Value is writable) THEN
 WRITE Present_Value = (a value x: x corresponds to a NORMAL state)
ELSE
 MAKE (Present_Value have a value x: x corresponds to a NORMAL state)
11. WAIT (Time_Delay)
12. BEFORE **Notification Fail Time**
 RECEIVE ConfirmedEventNotification-Request,
 'Process Identifier' = (any valid process ID),
 'Initiating Device Identifier' = IUT,
 'Event Object Identifier' = (the intrinsic reporting object being tested or ~~the object referenced by~~)

```

        'Time Stamp' = the Event Enrollment object being tested),
        'Notification Class' = (the current local time),
        'Priority' = (the configured notification class),
        'Event Type' = (the value configured to correspond to a TO-NORMAL transition),
        'Notify Type' = CHANGE_OF_STATE,
        'AckRequired' = EVENT | ALARM,
        'From State' = TRUE | FALSE,
        'To State' = OFFNORMAL,
        'Event Values' = NORMAL,
        Present_Value, Status_Flags
13. TRANSMIT BACnet-SimpleACK-PDU
14. IF (the object being tested is NOT an Event Enrollment object) THEN
    VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
15. VERIFY Event_State = NORMAL
16. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
    VERIFY Event_Time_Stamps = (the timestamp in step 45, *, the timestamp in step 112)
17. IF (the object being tested is a multi-state object that supports intrinsic reporting) THEN
18. IF (Present_Value is writable) THEN
    WRITE Present_Value = (a value x: x = one of the Fault_Values)
    ELSE
    MAKE (Present_Value have a value x: x = one of the Fault_Values)
19. WAIT (Time_Delay)
20. BEFORE Notification Fail Time
    RECEIVE ConfirmedEventNotification-Request,
        'Process Identifier' = (any valid process ID),
        'Initiating Device Identifier' = IUT,
        'Event Object Identifier' = (the intrinsic reporting object being tested),
        'Time Stamp' = (the current local time),
        'Notification Class' = (the configured notification class),
        'Priority' = (the value configured to correspond to a TO-FAULT transition),
        'Event Type' = CHANGE_OF_STATE,
        'Notify Type' = EVENT | ALARM,
        'AckRequired' = TRUE | FALSE,
        'From State' = NORMAL,
        'To State' = FAULT,
        'Event Values' = Present_Value, Status_Flags
21. TRANSMIT BACnet-SimpleACK-PDU
22. IF (the object being tested is NOT an Event Enrollment object) THEN
    VERIFY Status_Flags = (FALSE, TRUE, ?, ?)
23. VERIFY Event_State = FAULT
24. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
    VERIFY Event_Time_Stamps = (the timestamp in step 4, the timestamp in step 19, the timestamp in step
    11)
25. IF (the object being tested is a multi-state object that supports intrinsic reporting and Protocol_Revision is
    present and Protocol_Revision ≥ 1) THEN
    VERIFY Reliability = MULTI_STATE_FAULT
26. IF (Present_Value is writable) THEN
    WRITE Present_Value = (a value x: x corresponds to a NORMAL state)
    ELSE
    MAKE (Present_Value have a value x: x corresponds to a NORMAL state)
27. WAIT (Time_Delay)
28. BEFORE Notification Fail Time
    RECEIVE ConfirmedEventNotification-Request,
        'Process Identifier' = (any valid process ID),
        'Initiating Device Identifier' = IUT,
        'Event Object Identifier' = (the intrinsic reporting object being tested),

```

'Time Stamp' = (the current local time),
'Notification Class' = (the configured notification class),
'Priority' = (the value configured to correspond to a TO-NORMAL transition),
'Event Type' = CHANGE_OF_STATE,
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = FAULT,
'To State' = NORMAL,
'Event Values' = Present_Value, Status_Flags

29. TRANSMIT BACnet-SimpleACK-PDU
30. *IF (the object being tested is NOT an Event Enrollment object) THEN*
 VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
[End of the IF statement indention starting in line 2.]
31. VERIFY Event_State = NORMAL
32. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
 VERIFY Event_Time_Stamps = (the timestamp in step 4-5, the timestamp in step 4920, the timestamp in step 2628)

Notes to Tester: The 'Message Text' parameter is omitted in the test description because it is optional. The IUT may include this parameter in the notification messages. The time stamps indicated by "*" in steps 8-9 and 15-16 can have a value that indicates an unspecified time or a time that precedes the timestamp in step 4-5.

135.1-2009h-3 Change ConfirmedCOVNotification Service Initiation Tests to Non-infinite Lifetimes.

Rationale

This series of tests assumed the IUT supports infinite Lifetime (=0).

[Change multiple clauses under **Clause 8.2**, p. 107]

8.2 ConfirmedCOVNotification Service Initiation Tests

...

8.2.1 Change of Value Notification from an Analog Input, Analog Output, and Analog Value Object Present_Value Property

Purpose: To verify that the IUT can initiate ConfirmedCOVNotification service requests conveying a change of the Present_Value property of Analog Input, Analog Output, and Analog Value objects.

Test Concept: A subscription for COV notifications is established, *using a Lifetime of L. L shall be set to a value less than 24 hours and large enough to complete the test.* The Present_Value of the monitored object is changed by an amount less than the COV increment and it is verified that no COV notification is received. The Present_Value is then changed by an amount greater than the COV increment and a notification shall be received. The Present_Value may be changed using the WriteProperty service or by another means such as changing the input signal represented by an Analog Input object. For some implementations it may be necessary to write to the Out_Of_Service property first to accomplish this task. For implementations where it is not possible to write to these properties at all the vendor shall provide an alternative trigger mechanism to accomplish this task. All of these methods are equally acceptable.

Configuration Requirements: At the beginning of the test, the Out_Of_Service property shall have a value of FALSE.

Test Steps:

REPEAT X = (one supported object of each type from the set Analog Input, Analog Output, and Analog Value) DO {

1. TRANSMIT SubscribeCOV-Request,
 - 'Subscriber Process Identifier' = (any value > 0 chosen by the TD),
 - 'Monitored Object Identifier' = X,
 - 'Issue Confirmed Notifications' = TRUE,
 - 'Lifetime' = θL
2. RECEIVE BACnet-SimpleACK-PDU
3. **BEFORE Notification Fail Time**
 - RECEIVE ConfirmedCOVNotification-Request,
 - 'Subscriber Process Identifier' = (the same value used in step 1),
 - 'Initiating Device Identifier' = IUT,
 - 'Monitored Object Identifier' = X,
 - 'Time Remaining' = θ (any value appropriate for the Lifetime selected),
 - 'List of Values' = (the initial Present_Value and initial Status_Flags)
4. TRANSMIT BACnet-SimpleACK-PDU
5. TRANSMIT ReadProperty-Request,
 - 'Object Identifier' = X,
 - 'Property Identifier' = COV_Increment
6. RECEIVE BACnet-ComplexACK-PDU,
 - 'Object Identifier' = X,
 - 'Property Identifier' = COV_Increment,
 - 'Property Value' = (a value "increment" that will be used below)
7. IF (Out_Of_Service is writable) THEN
 - WRITE X, Out_Of_Service = TRUE
 - RECEIVE BACnet-SimpleACK-PDU
 - BEFORE Notification Fail Time**
 - RECEIVE ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = (the same value used in step 1),
'Initiating Device Identifier' = IUT,
'Monitored Object Identifier' = X,
'Time Remaining' = θ (any value appropriate for the Lifetime selected),
'List of Values' = (the initial Present_Value and new Status_Flags)

TRANSMIT BACnet-SimpleACK-PDU

8. IF (Present_Value is now writable) THEN
 WRITE X, Present_Value = (any value that differs from "initial Present_Value" by less than "increment")
 RECEIVE BACnet-SimpleACK-PDU
ELSE
 MAKE (Present_Value = any value that differs from "initial Present_Value" by less than "increment")

9. WAIT **NotificationFailTime**

10. CHECK (verify that no COV notification was transmitted)

11. IF (Present_Value is now writable) THEN
 WRITE X, Present_Value = (any value that differs from "initial Present_Value" by an amount greater than "increment")
 RECEIVE BACnet-SimpleACK-PDU
ELSE
 MAKE (Present_Value = any value that differs from "initial Present_Value" by an amount greater than "increment")

12. BEFORE **NotificationFailTime**
 RECEIVE ConfirmedCOVNotification-Request,
 'Subscriber Process Identifier' = (the same value used in step 1),
 'Initiating Device Identifier' = IUT,
 'Monitored Object Identifier' = X,
 'Time Remaining' = θ (any value appropriate for the Lifetime selected),
 'List of Values' = (the new Present_Value and new Status_Flags)

13. TRANSMIT BACnet-SimpleACK-PDU

14. TRANSMIT SubscribeCOV-Request,
 'Subscriber Process Identifier' = (the same value used in step 1),
 'Monitored Object Identifier' = X

15. RECEIVE BACnet-SimpleACK-PDU

16. IF (Out_Of_Service is writable) THEN
 WRITE X, Out_Of_Service = FALSE
 RECEIVE BACnet-SimpleACK-PDU

[NOTE: The same changes made in the above test shall be repeated for the remaining tests in this section.]

8.2.2 Change of Value Notification from an Analog Input, Analog Output, and Analog Value Object Status_Flags Property

[Make same changes to test as shown in test 8.2.1 above]

8.2.3 Change of Value Notification from a Binary Input, Binary Output, and Binary Value Object Present_Value Property

[Make same changes to test as shown in test 8.2.1 above]

8.2.4 Change of Value Notification from a Binary Input, Binary Output, and Binary Value Object Status_Flags Property

[Make same changes to test as shown in test 8.2.1 above]

8.2.5 Change of Value Notification from a Multi-state Input, Multi-state Output, Multi-state Value, Life Safety Point, or Life Safety Zone Object Present_Value Property

[Make same changes to test as shown in test 8.2.1 above]

8.2.6 Change of Value Notification from a Multi-state Input, Multi-state Output Multi-state Value, Life Safety Point, and Life Safety Zone Object Status_Flags Property

[Make same changes to test as shown in test 8.2.1 above]

8.2.7 Change of Value Notification from Loop Object Present_Value Property

[Make same changes to test as shown in test 8.2.1 above]

8.2.8 Change of Value Notification from a Loop Object Status_Flags Property

[Make same changes to test as shown in test 8.2.1 above]

135.1-2009h-4 Change intrinsic tests for Event Enrollment Objects.

Rationale
A number of tests were expecting an incorrect value for the 'Event Object Identifier' field.

[Change **Clause 8.4.4** steps 5 & 12, p. 124
Clause 8.4.6 steps 8, 19, 30, 41, p. 129
Clause 8.4.7 steps 3, & 6, p. 132]

'Event Object Identifier' = (the intrinsic reporting object being tested or ~~the object referenced by the~~ Event Enrollment object being tested),

[Change **Clause 8.4.5** steps 8, 19, p. 126]

'Event Object Identifier' = (the Loop object being tested or ~~the object referenced by the~~ Event Enrollment object being tested),

[Add a new entry to **History of Revisions**, p. 489]

(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>Summary of Changes to the Standard</i>
...
Addendum h to ANSI/ASHRAE 135.1-2009 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.
<ol style="list-style-type: none">1. Change GetEventInformation Chaining Tests.2. Change CHANGE_OF_STATE Test for Event Enrollment Object.3. Change ConfirmedCOVEventNotification Service Initiation Tests to Non-infinite Lifetimes.4. Change intrinsic tests for Event Enrollment Object.

**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.

