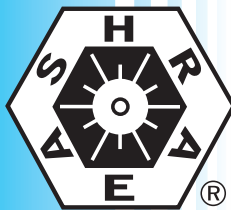


ANSI/ASHRAE Addendum c to  
ANSI/ASHRAE Standard 135.1-2003



# ASHRAE STANDARD

## Method of Test for Conformance to BACnet®

Approved by the ASHRAE Standards Committee on June 23, 2007; by the ASHRAE Board of Directors on June 27, 2007; and by the American National Standards Institute on June 28, 2007.

ASHRAE Standards are scheduled to be updated on a five-year cycle; the date following the standard number is the year of ASHRAE Board of Directors approval. The latest copies may be purchased 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).

© Copyright 2007 ASHRAE, 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: Frank E. Jacob**

William O. Swan, III, <i>Chair*</i>	Steven T. Bushby	James G. Luth*
David Robin, <i>Vice-Chair</i>	Sharon E. Dinges*	John J. Lynch
Carl Neilson, <i>Secretary</i>	Craig P. Gemmill	Jerald P. Martocci
Donald P. Alexander*	David G. Holmberg*	Carl J. Ruther*
Barry B. Bridges*	Robert L. Johnson*	David G. Shike*
Coleman L. Brumley, Jr.*	Stephen Karg	David B. Thompson*
Ernest C. Bryant*	J. Damian Ljungquist*	J. Michael Whitcomb
James F. Butler		

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

---

**ASHRAE STANDARDS COMMITTEE 2006–2007**

David E. Knebel, <i>Chair</i>	James D. Lutz
Stephen D. Kennedy, <i>Vice-Chair</i>	Carol E. Marriott
Michael F. Beda	Merle F. McBride
Donald L. Brandt	Mark P. Modera
Steven T. Bushby	Ross D. Montgomery
Paul W. Cabot	H. Michael Newman
Hugh F. Crowther	Stephen V. Santoro
Samuel D. Cummings, Jr.	Lawrence J. Schoen
Robert G. Doerr	Stephen V. Skalko
Roger L. Hedrick	Bodh R. Subherwal
John F. Hogan	Jerry W. White, Jr.
Eli P. Howard, III	James E. Woods
Frank E. Jakob	Richard D. Hermans, <i>BOD ExO</i>
Jay A. Kohler	Hugh D. McMillan, III, <i>CO</i>

Claire B. Ramspeck, *Assistant Director of Technology for Standards and Special Projects*

---

**SPECIAL NOTE**

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

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

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

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

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

**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*c* to ANSI/ASHRAE Standard 135.1-2003 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.

- 135.1-2003*c*-1. Update references to refer to 135-2004, p. 1.
- 135.1-2003*c*-2. Add new object types from 135-2004, p. 22.
- 135.1-2003*c*-3. Omit certain tests based on Protocol\_Revision, p. 26.
- 135.1-2003*c*-4. Exception schedule priority requirements, p. 29.
- 135.1-2003*c*-5. Minor corrections, p. 30.

SSPC 135 wishes to recognize the efforts of the following people in developing this addendum: Sharon Dinges, Howard Coleman, and John Hartman. The committee is also grateful to H. Michael Newman.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135.1-2003 and addenda is indicated through the use of *italics*, while deletions are indicated by ~~striketrough~~. Where entirely new subclauses are added, plain type is used throughout.

## 135.1-2003c-1. Update references to correspond to 135-2004

### Rationale

135.1-2003 makes references to 135-2001 that must be updated to refer to 135-2004.

### Addendum 135.1-2003c-1

[Change 3, p. 1]

#### 3. DEFINITIONS

All definitions from ANSI/ASHRAE Standard ~~135-2001~~ 135-2004 also apply to this addendum.

[Change 6.2.8, p. 25]

#### 6.2.8 WAIT Statement

The WAIT statement is used to pause the execution of the TD for some specified amount of time.

<wait statement> ::= WAIT <timer value>

Test Steps: The TD shall pause the amount of time specified by the <timer value> before proceeding to the next test step. The <timer value> shall be one of the timers specified in 6.3 of this standard, in ANSI/ASHRAE ~~135-2001~~, 135-2004, or as otherwise specified.

[Change 6.4, p. 27]

#### 6.4 BACnet References

All references to BACnet clauses in this standard refer to ANSI/ASHRAE ~~135-2001~~. 135-2004, *except when otherwise noted ("BACnet-2001" refers to ANSI/ASHRAE 135-2001).*

[Change 7.3.1.1, p. 29]

#### 7.3.1.1 Out\_Of\_Service, Status\_Flags, and Reliability Tests

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: 12.1.7, 12.1.9, 12.1.10, 12.2.7, 12.2.9, 12.2.10, ~~12.3.6, 12.3.8, 12.3.9~~, 12.3.7, 12.3.9, 12.3.10, 12.4.6, 12.4.8, 12.4.9, ~~12.5.7, 12.5.9, 12.5.10~~, 12.6.7, 12.6.9, 12.6.10, ~~12.7.6, 12.7.8, 12.7.9~~, 12.7.7, 12.7.9, 12.7.10, 12.8.6, 12.8.8, 12.8.9, ~~12.14.8, 12.14.10, 12.14.11~~, 12.15.8, 12.15.10, 12.15.11, ~~12.16.6, 12.16.8, 12.16.9~~, 12.16.8, 12.16.10, 12.16.11, ~~12.17.7, 12.17.9, 12.17.10~~, 12.17.6, 12.17.8, 12.17.9, 12.18.7, 12.18.9, 12.18.10, ~~12.19.6, 12.19.8, and 12.19.9~~, 12.19.7, 12.19.9, 12.19.10, 12.20.6, 12.20.8, 12.20.9, 12.23.7, 12.23.9, and 12.23.10.

Purpose: This test case verifies that Present\_Value is writable when Out\_Of\_Service is TRUE. It also verifies the interrelationship between the Out\_Of\_Service, Status\_Flags, and Reliability properties. If the PICS indicates that the Out\_Of\_Service property of the object under test is not writable, and if the value of the property cannot be changed by other means, then this test shall be omitted. This test applies to *Accumulator*, Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value, Life Safety Point, Life Safety Zone, Multi-state Input, Multi-state Output, Multi-state Value, ~~and~~ Loop *and Pulse Converter* objects.

[Change 7.3.1.2, p. 30]

#### **7.3.1.2 Relinquish Default Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.2.16, 12.3.12, 12.6.22, 12.7.20, 12.18.14 and 12.19.3.~~ *12.3.16, 12.4.12, 12.7.22, 12.8.20, 12.19.14, and 12.20.13.*

[Change 7.3.1.4, p. 32]

#### **7.3.1.4 Minimum\_Off\_Time**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.6.19, 12.7.17,~~ *12.7.19, 12.8.17,* 19.2.3, and Annex I.

[Change 7.3.1.5, p. 32]

#### **7.3.1.5 Minimum\_On\_Time**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.6.20, 12.7.18,~~ *12.7.20, 12.8.18,* 19.2.3, and Annex I.

[Change 7.3.1.6, p. 33]

#### **7.3.1.6 Override of Minimum Time**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~19.~~ *19.2.*

[Change 7.3.1.8, p. 33]

#### **7.3.1.8 Binary Object Change of State Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.4.14, 12.4.15, 12.4.16, 12.5.14, 12.5.15, 12.5.16, 12.6.14, 12.6.15, and 12.6.16.~~ *12.6.14, 12.6.15, 12.6.16, 12.7.14, 12.7.15, 12.7.16, 12.8.12, 12.8.13, and 12.8.14.*

[Change 7.3.1.9, p. 35]

#### **7.3.1.9 Binary Object Elapsed Active Time Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.5.17, 12.5.18, 12.6.17, 12.6.18, 12.7.15, and 12.7.16.~~ *12.6.17, 12.6.18, 12.7.17, 12.7.18, 12.8.15, and 12.8.16.*

[Change 7.3.1.10, p. 36]

### 7.3.1.10 Event\_Enable Tests

Dependencies: ConfirmedEventNotification Service Initiation Tests, 8.4; UnconfirmedEventNotification Service Initiation Tests, 8.5; ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: *12.1.27, ~~12.1.23~~, ~~12.2.24~~, ~~12.3.20~~, ~~12.5.22~~, ~~12.6.26~~, ~~12.7.24~~, ~~12.11.10~~, ~~12.2.23~~, ~~12.3.24~~, ~~12.4.20~~, ~~12.6.22~~, ~~12.7.26~~, ~~12.8.24~~, ~~12.12.10~~, ~~12.14.18~~, ~~12.15.18~~, ~~12.15.19~~, ~~12.16.19~~, ~~12.16.33~~, ~~12.17.17~~, ~~12.18.18~~, ~~12.19.18~~, ~~12.17.33~~, ~~12.18.17~~, ~~12.19.18~~, ~~12.20.18~~, ~~12.23.26~~, and ~~12.23.23~~. 12.25.22.*

Purpose: To verify that notification messages are transmitted only if the bit in Event\_Enable corresponding to the event transition has a value of TRUE. This test applies to Event Enrollment objects and *Accumulator*, Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value, Life Safety Point, Life Safety Zone, Loop, Multi-state Input, Multi-state Output, ~~and Multi-state Value Value~~, *Pulse Converter and Trend Log* objects that support intrinsic reporting.

[Change 7.3.1.11, p. 38]

### 7.3.1.11 Acked\_Transitions Tests

Dependencies: ConfirmedEventNotification Service Initiation Tests, 8.4; UnconfirmedEventNotification Service Initiation Tests, 8.5; AcknowledgeAlarm Service Execution Tests, 9.1; ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: *12.1.28, ~~12.1.24~~, ~~12.2.25~~, ~~12.3.21~~, ~~12.5.23~~, ~~12.6.27~~, ~~12.7.25~~, ~~12.11.11~~, ~~12.2.24~~, ~~12.3.25~~, ~~12.4.21~~, ~~12.6.23~~, ~~12.7.27~~, ~~12.8.25~~, ~~12.12.11~~, ~~12.14.19~~, ~~12.15.19~~, ~~12.15.20~~, ~~12.16.20~~, ~~12.16.34~~, ~~12.17.18~~, ~~12.18.19~~, ~~12.19.19~~, ~~12.17.34~~, ~~12.18.18~~, ~~12.19.19~~, ~~12.20.19~~, ~~12.23.27~~ and ~~12.23.24~~. 12.25.23.*

Purpose: To verify that the Acked\_Transitions property tracks whether or not an acknowledgment has been received for a previously issued event notification. It also verifies the interrelationship between Status\_Flags and Event\_State. This test applies to Event Enrollment objects and *Accumulator*, Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value, Life Safety Point, Life Safety Zone, Loop, Multi-state Input, Multi-state Output, ~~and Multi-state Value Value~~, *Pulse Converter and Trend Log* objects that support intrinsic reporting.

[Change 7.3.1.12, p. 42]

### 7.3.1.12 Notify\_Type Test

Dependencies: ConfirmedEventNotification Service Initiation Tests, 8.4; UnconfirmedEventNotification Service Initiation Tests, 8.5; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: *12.1.29, ~~12.1.25~~, ~~12.2.26~~, ~~12.3.22~~, ~~12.5.24~~, ~~12.6.28~~, ~~12.7.26~~, ~~12.11.6~~, ~~12.2.25~~, ~~12.3.26~~, ~~12.4.22~~, ~~12.6.24~~, ~~12.7.28~~, ~~12.8.26~~, ~~12.12.6~~, ~~12.14.20~~, ~~12.15.20~~, ~~12.15.21~~, ~~12.16.21~~, ~~12.16.35~~, ~~12.17.19~~, ~~12.18.20~~, ~~12.19.20~~, ~~12.17.35~~, ~~12.18.19~~, ~~12.19.20~~, ~~12.20.20~~, ~~12.23.28~~, and ~~12.23.25~~. 12.25.24.*

Purpose: To verify that the value of the Notify\_Type property determines whether an event notification is transmitted as an alarm or as an event. This test applies to Event Enrollment objects and *Accumulator*, Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value, Life Safety Point, Life Safety Zone, Loop, Multi-state Input, Multi-state Output, ~~and Multi-state Value Value~~, *Pulse Converter and Trend Log* objects that support intrinsic reporting.

[Change 7.3.1.13, p. 43]

#### **7.3.1.13 Limit\_Enable Test**

Dependencies: ConfirmedEventNotification Service Initiation Tests, 8.4; UnconfirmedEventNotification Service Initiation Tests, 8.5; ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.1.22, 12.2.23, and 12.3.19.~~ 12.1.26, 12.2.22, 12.3.23, 12.4.19, and 12.23.25.

Purpose: To verify that the Limit\_Enable property correctly enables or disables reporting of out of range events. This test applies to *Accumulator*, Analog Input, Analog Output, ~~and Analog Value Value~~, and *Pulse Converter* objects that support intrinsic reporting. If the Limit\_Enable property is not writable and cannot be reconfigured this test shall be omitted.

[Change 7.3.1.14.1, p. 46]

#### **7.3.1.14.1 Process\_Identifier Property Test**

Dependencies: None.

~~BACnet~~ BACnet-2001 Reference Clause: 12.11.4.

Purpose: To verify that the Process\_Identifier property correctly supports the required value range. This test shall be performed only if the Protocol\_Revision property is present in the Device object and has a value ~~greater than or equal to 4.~~ *in the range of 1 through 3.*

[Change 7.3.1.14.2, p. 46]

#### **7.3.1.14.2 Recipient\_List Test**

Dependencies: None

BACnet Reference Clause: ~~12.20.8.~~ 12.21.8.

[Change 7.3.2.1.1, p. 47]

#### **7.3.2.1.1 Input Tracking Test**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clause: ~~12.1.4.~~ 12.2.4.

[Change 7.3.2.2.1, p. 47]

#### **7.3.2.2.1 Output Tracking Test**

Dependencies: WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.2.4.~~ 12.3.4.

[Change 7.3.2.4, p. 48]

#### **7.3.2.4 Averaging Object Tests**

An Averaging object provides a way to monitor the average, minimum, and maximum values attained by a sampled property. The datatype of the sampled property can be BOOLEAN, INTEGER, Unsigned, Enumerated, or Real. The tests in this clause shall be repeated once for each of these datatypes.

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.4.~~ 12.5.

[Change 7.3.2.4.2, p. 50]

#### **7.3.2.4.2 Managing the Sample Window**

BACnet Reference Clause: ~~12.4.15~~ 12.5.15.

[Change 7.3.2.5.1, p. 50]

#### **7.3.2.5.1 Input Tracking Test**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clause: ~~12.5.4.~~ 12.6.4.

[Change 7.3.2.5.3, p. 51]

#### **7.3.2.5.3 Polarity Property Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.5.11.~~ 12.6.11.

[Change 7.3.2.6.1, p. 52]

#### **7.3.2.6.1 Output Tracking Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.6.4.~~ 12.7.4.

[Change 7.3.2.6.3, p. 52]

#### **7.3.2.6.3 Polarity Property Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.6.11.~~ 12.7.11.

[Change 7.3.2.8, p. 53]

#### **7.3.2.8 Calendar Test**

These tests verify that the Present\_Value property of the Calendar object bears the relationship to Date\_List specified by BACnet Clause ~~12.8.6.~~ 12.9.6.



[Change **7.3.2.8.1**, p. 53]

**7.3.2.8.1 Single Date Rollover Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.8.~~ 12.9.

[Change **7.3.2.8.2**, p. 54]

**7.3.2.8.2 Date Range Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.8.~~ 12.9.

[Change **7.3.2.8.3**, p. 54]

**7.3.2.8.3 WeekNDay Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.8.~~ 12.9.

[Change **7.3.2.9.1**, p. 55]

**7.3.2.9.1 All Writes Successful with Post Delay Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.9.8.~~ 12.10.8.

[Change **7.3.2.9.2**, p. 56]

**7.3.2.9.2 Quit on Failure Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.9.8.~~ 12.10.8.

[Change **7.3.2.9.3**, p. 56]

**7.3.2.9.3 External Writes Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.9.8.~~ 12.10.8.

[Change **7.3.2.9.4**, p. 57]

**7.3.2.9.4 Empty Action List Test**

Dependencies: WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.9.8.~~ 12.10.8.

[Change 7.3.2.9.5, p. 57]

**7.3.2.9.5 Action 0 Test**

Dependencies: WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.9.8~~; 12.10.8.

[Change 7.3.2.9.6, p. 57]

**7.3.2.9.6 Action\_Text Test**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clauses: ~~12.9.8 and 12.9.9~~; 12.10.8 and 12.10.9.

[Change 7.3.2.9.7, p. 58]

**7.3.2.9.7 Write While In\_Process is TRUE Test.**

Dependencies: WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.9.8 and 12.9.9~~; 12.10.8 and 12.10.9.

[Change 7.3.2.13, p. 58]

**7.3.2.13 Group Object Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.13~~; 12.14.

[Change 7.3.2.14.1, p. 59]

**7.3.2.14.1 Tracking Value Test**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clause: ~~12.14~~; 12.15.5.

[Change 7.3.2.15.1, p. 60]

**7.3.2.15.1 Tracking Value Test**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clause: ~~12.15~~; 12.16.5.

[Change 7.3.2.16.1, p. 60]

**7.3.2.16.1 Manipulated\_Variable\_Reference Tracking**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clause: ~~12.16.12~~; 12.17.12.

[Change 7.3.2.16.2, p. 61]

**7.3.2.16.2 Controlled\_Variable\_Reference Tracking**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clause: ~~12.16.13~~. 12.17.13.

[Change 7.3.2.16.3, p. 62]

**7.3.2.16.3 Setpoint\_Reference Tracking**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clause: ~~12.16.16~~. 12.17.16.

[Change 7.3.2.17.2, p. 62]

**7.3.2.17.2 Number\_Of\_States and State\_Text**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clauses: ~~12.17.11 and 12.17.12~~. 12.18.11 and 12.18.12.

[Change 7.3.2.17.4, p. 63]

**7.3.2.17.4 Input Tracking Test**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clause: ~~12.17.4~~. 12.18.4.

[Change 7.3.2.18.2, p. 63]

**7.3.2.18.2 Number\_Of\_States and State\_Text**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clauses: ~~12.18.11 and 12.18.12~~. 12.19.11 and 12.19.12.

[Change 7.3.2.18.5, p. 64]

**7.3.2.18.5 Output Tracking Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.18.4~~. 12.19.4.

[Change 7.3.2.19.2, p. 64]

**7.3.2.19.2 Number\_Of\_States and State\_Text**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clause: ~~12.19~~. 12.20.10 and 12.20.11.

[Change 7.3.2.20.1, pp. 64-65]

#### **7.3.2.20.1 Priority Tests**

Dependencies: ConfirmedEventNotification Service Initiation Tests, 8.4; UnconfirmedEventNotification Service Initiation Tests, 8.5; ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.20.6.~~ 12.21.6.

[Change 7.3.2.20.2.1, p. 67]

#### **7.3.2.20.2.1 Ack\_Required False Test**

Dependencies: ConfirmedEventNotification Service Initiation Tests, 8.4; UnconfirmedEventNotification Service Initiation Tests, 8.5; ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.20.7.~~ 12.21.7.

[Change 7.3.2.20.2.2, p. 67]

#### **7.3.2.10.2.2 Ack\_Required True Test**

Dependencies: ConfirmedEventNotification Service Initiation Tests, 8.4; UnconfirmedEventNotification Service Initiation Tests, 8.5; ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.20.7.~~ 12.21.7.

[Change 7.3.2.20.3.1, p. 68]

#### **7.3.2.20.3.1 ValidDays Test**

Dependencies: ConfirmedEventNotification Service Initiation Tests, 8.4; UnconfirmedEventNotification Service Initiation Tests, 8.5; ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.20.8.~~ 12.21.8.

[Change 7.3.2.20.3.2, p. 69]

#### **7.3.2.20.3.2 FromTime and ToTime Test**

Dependencies: ValidDays Test, 7.3.2.20.3.1; ConfirmedEventNotification Service Initiation Tests, 8.4; UnconfirmedEventNotification Service Initiation Tests, 8.5; ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.20.8.~~ 12.21.8.

[Change 7.3.2.20.3.3, p. 69]

#### **7.3.2.20.3.3 IssueConfirmedNotifications Test**

Dependencies: ConfirmedEventNotification Service Initiation Tests, 8.4; UnconfirmedEventNotification Service Initiation Tests, 8.5; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.20.8.~~ 12.21.8.

[Change 7.3.2.20.3.4, p. 71]

#### **7.3.2.20.3.4 Transitions Test**

Dependencies: ConfirmedEventNotification Service Initiation Tests, 8.4; UnconfirmedEventNotification Service Initiation Tests, 8.5; ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.20.8~~. 12.21.8.

[Change 7.3.2.22.1, p. 73]

#### **7.3.2.22.1 Effective\_Period Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.6~~. 12.24.6.

[Change 7.3.2.22.2, p. 74]

#### **7.3.2.22.2 Weekly\_Schedule Property Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.7~~. 12.24.7.

[Change 7.3.2.22.3.1, p. 75]

#### **7.3.2.22.3.1 Calendar Reference Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.8~~. 12.24.8.

[Change 7.3.2.22.3.2, p. 76]

#### **7.3.2.22.3.2 Calendar Entry Date Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.8~~. 12.24.8.

[Change 7.3.2.22.3.3, p. 76]

#### **7.3.2.22.3.3 Calendar Entry DateRange Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.8~~. 12.24.8.

[Change 7.3.2.22.3.4, p. 77]

#### **7.3.2.22.3.4 Calendar Entry WeekNDay Month Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.8~~; 12.24.8.

[Change 7.3.2.22.3.5, p. 78]

**7.3.2.22.3.5 Calendar Entry WeekNDay Week Of Month Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.8~~; 12.24.8.

[Change 7.3.2.22.3.6, p. 78]

**7.3.2.22.3.6 Calendar Entry WeekNDay Last Week Of Month Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.8~~; 12.24.8.

[Change 7.3.2.22.3.7, p. 79]

**7.3.2.22.3.7 Calendar Entry WeekNDay Day Of Week Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.8~~; 12.24.8.

[Change 7.3.2.22.3.8, p. 80]

**7.3.2.22.3.8 Event Priority Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.8~~; 12.24.8.

[Change 7.3.2.22.3.9, p. 80]

**7.3.2.22.3.9 List of BACnetTimeValue Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.8~~; 12.24.8.

[Change 7.3.2.22.4, p. 81]

**7.3.2.22.4 Weekly\_Schedule and Exception\_Schedule Interaction Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clauses: ~~12.22.7, 12.22.8~~; 12.24.7, 12.24.8.

[Change 7.3.2.22.5, p. 81]

#### **7.3.2.22.5 Exception\_Schedule Restoration Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; ReinitializeDevice Service Execution Tests, 9.27; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clauses: ~~12.22.7, 12.22.8~~. 12.24.7, 12.24.8.

[Change 7.3.2.22.6, p. 82]

#### **7.3.2.22.6 Weekly\_Schedule Restoration Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; ReinitializeDevice Service Execution Tests, 9.27; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.7~~. 12.24.7.

[Change 7.3.2.22.7, p. 83]

#### **7.3.2.22.7 List\_Of\_Object\_Property\_Reference Internal Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.9~~. 12.24.10.

[Change 7.3.2.22.8, p. 83]

#### **7.3.2.22.8 List\_Of\_Object\_Property\_Reference External Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30.

BACnet Reference Clause: ~~12.22.9~~. 12.24.10.

[Change 7.3.2.23.1, p. 84]

#### **7.3.2.23.1 Log\_Enable Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.23.5~~. 12.25.5.

[Change 7.3.2.23.2, p. 85]

#### **7.3.2.23.2 Start\_Time Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.23.6~~. 12.25.6.

[Change 7.3.2.23.3, p. 86]

**7.3.2.23.3 Stop\_Time Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.23.7~~ 12.25.7.

[Change 7.3.2.23.4, pp. 86-87]

**7.3.2.23.4 Log\_Interval Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.23.9~~ 12.25.9.

[Change 7.3.2.23.5, p. 87]

**7.3.2.23.5 COV\_Resubscription\_Interval Test**

Dependencies: Confirmed Notifications Subscription, 8.10.1; Unconfirmed Notifications Subscription, 8.10.2.

BACnet Reference Clause: ~~12.23.10~~ 12.25.10.

[Change 7.3.2.23.6.1, p. 88]

**7.3.2.23.6.1 Stop\_When\_Full TRUE Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.23.12~~ 12.25.12.

[Change 7.3.2.23.6.2, p. 89]

**7.3.2.23.6.2 Stop\_When\_Full FALSE Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.23.12~~ 12.25.12.

[Change 7.3.2.23.7, p. 89]

**7.3.2.23.7 Buffer\_Size Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.23.13~~ 12.25.13.

[Change 7.3.2.23.8, p. 90]

**7.3.2.23.8 Record\_Count Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.23.15~~ 12.25.15.



[Change 7.3.2.23.9, p. 90]

#### **7.3.2.23.9 Total\_Record\_Count Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.23.16~~. 12.25.16.

Purpose: To verify that the Total\_Record\_Count property increments for each record added to the Log\_Buffer, even after Buffer\_Size records have been added. (Note: it is not reasonable to test for the requirement of BACnet Clause ~~12.23.16~~ 12.25.16 that the value wrap from -1 to 0; even if a record was collected every 100<sup>th</sup> of a second it could take more than 497 days to complete the test.)

[Change 7.3.2.23.10, p. 91]

#### **7.3.2.23.10 Notification\_Threshold Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~12.23.17~~. 12.25.17.

[Change 7.3.2.23.11, p. 92]

#### **7.3.2.23.11 Notification Time Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet-2001 Reference Clauses: 12.23.19, 12.23.20, and 12.23.26.

[Change 7.3.2.23.12, p. 94]

#### **7.3.2.23.12 COV Subscription Failure Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.23.5, 12.23.9, and 12.23.10~~. 12.25.5, 12.25.9, and 12.25.10.

[Change 8.3, p. 105]

### **8.3 UnconfirmedCOVNotification Service Initiation Tests**

This clause defines the tests necessary to demonstrate support for initiating UnconfirmedCOVNotification service requests. The UnconfirmedCOVNotification tests are specific to a particular object type that provides intrinsic COV reporting capabilities. The IUT shall pass all of the tests for each object type that is claimed to be supported in the PICS.

Dependencies: SubscribeCOV Service Execution Tests, 9.10; ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: ~~13.14~~. 13.7.

[Change 8.4.1, p. 107]

#### **8.4.1 CHANGE\_OF\_BITSTRING Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11~~, 12.12.5, 12.12.7, 13.3.1, and 13.8.

[Change 8.4.2, p. 108]

#### **8.4.2 CHANGE\_OF\_STATE Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.5, 12.7, 12.17, 12.19~~, 12.6, 12.8, 12.18, 12.20, 13.2, 13.3.2, and 13.8.

[Change 8.4.3.1, p. 111]

#### **8.4.3.1 Numerical Algorithm**

The test in this clause applies to use of the CHANGE\_OF\_VALUE algorithm applied to Integer or Real datatypes.

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11~~, 12.12, 13.3.2, and 13.8.

[Change 8.4.3.2, p. 112]

#### **8.4.3.2 Bitstring Algorithm**

The test in this clause applies to use of the CHANGE\_OF\_VALUE algorithm applied to Bitstring datatypes.

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11~~, 12.12, 13.3.2, and 13.8.

[Change 8.4.4, p. 113]

#### **8.4.4 COMMAND\_FAILURE Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.6, 12.18~~, 12.7, 12.12, 12.19, 13.2, 13.3.4, and 13.8.

[Change 8.4.5, p. 115]

#### **8.4.5 FLOATING\_LIMIT Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11, 12.16, 12.20~~, 12.12, 12.17, 12.21, 13.2, 13.3.5, and 13.8.

[Change 8.4.6, p. 118]

#### **8.4.6 OUT\_OF\_RANGE Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.1, 12.2, 12.3, 12.11~~, 12.2, 12.3, 12.4, 12.12, 12.23, 13.2, 13.3.6, and 13.8.

[Change **8.4.7**, p. 121]

#### **8.4.7 BUFFER\_READY Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clauses: ~~12.11, 12.23~~, 12.12, 12.25, 13.2, 13.3.7, and 13.8.

[Change **8.4.8.1**, p. 122]

#### **8.4.8.1 NORMAL to OFFNORMAL Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change **8.4.8.2**, p. 123]

#### **8.4.8.2 OFFNORMAL to NORMAL Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change **8.4.8.3**, p. 124]

#### **8.4.8.3 NORMAL to LIFE\_SAFETY\_ALARM Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change **8.4.8.4**, p. 125]

#### **8.4.8.4 LIFE\_SAFETY\_ALARM to NORMAL Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change **8.4.8.5**, p. 126]

#### **8.4.8.5 LIFE\_SAFETY\_ALARM to OFFNORMAL Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change **8.4.8.6**, p. 127]

#### **8.4.8.6 OFFNORMAL to LIFE\_SAFETY\_ALARM Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change **8.4.8.7**, p. 128]

**8.4.8.7 Mode Transition Tests when Event State is Maintained**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change **8.4.8.8**, p. 130]

**8.4.8.8 NORMAL to OFFNORMAL Mode Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change **8.4.8.9**, p. 131]

**8.4.8.9 OFFNORMAL to NORMAL Mode Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change **8.4.8.10**, p. 132]

**8.4.8.10 NORMAL to LIFE\_SAFETY\_ALARM Mode Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change **8.4.8.11**, p. 133]

**8.4.8.11 LIFE\_SAFETY\_ALARM to NORMAL Mode Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change **8.4.8.12**, p. 134]

**8.4.8.12 LIFE\_SAFETY\_ALARM to OFFNORMAL Mode Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change **8.4.8.13**, p. 135]

**8.4.8.13 OFFNORMAL to LIFE\_SAFETY\_ALARM Mode Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change 8.4.8.14, p. 137]

#### **8.4.8.14 TO-FAULT Transition Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.8 and Figure ~~13.9~~ 13-9.

[Change 8.5.1, p. 138]

#### **8.5.1 CHANGE\_OF\_BITSTRING Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11~~, 12.12, 13.3.1, and ~~13.7~~ 13.9.

[Change 8.5.2, p. 138]

#### **8.5.2 CHANGE\_OF\_STATE Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.5, 12.7, 12.17, 12.19~~, 12.6, 12.8, 12.18, 12.20, 13.2, 13.3.2, and 13.9.

[Change 8.5.3.1, p. 139]

#### **8.5.3.1 Numerical Algorithm**

The test in this clause applies to use of the CHANGE\_OF\_VALUE algorithm applied to Integer or Real datatypes.

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11~~, 12.12, 13.3.3, and 13.9.

[Change 8.5.3.2, p. 139]

#### **8.5.3.2 Bitstring Algorithm**

The test in this clause applies to use of the CHANGE\_OF\_VALUE algorithm applied to Bitstring datatypes.

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11~~, 12.12, 13.3.3, and 13.9.

[Change 8.5.4, pp. 130-140]

#### **8.5.4 COMMAND\_FAILURE Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.6, 12.18~~, 12.7, 12.12, 12.19, 13.2, 13.3.4, and 13.9.

[Change 8.5.5, p. 140]

#### **8.5.5 FLOATING\_LIMIT Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11, 12.16,~~ 12.12, 12.17, 13.2, 13.3.5, and 13.9.

[Change 8.5.6, p. 140]

#### **8.5.6 OUT\_OF\_RANGE Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.1, 12.2, 12.3, 12.11,~~ 12.2, 12.3, 12.4, 12.12, 13.2, 13.3.6, and 13.9.

[Change 8.5.7, p. 141]

#### **8.5.7 BUFFER\_READY Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18.

BACnet Reference Clauses: ~~12.11, 12.23,~~ 12.12, 12.25, 13.2, 13.3.7, and 13.9.

[Change 8.5.8.1, p. 141]

#### **8.5.8.1 NORMAL to OFFNORMAL Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15,~~ 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9.~~ 13-9.

[Change 8.5.8.2, p. 141]

#### **8.5.8.2 OFFNORMAL to NORMAL Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15,~~ 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9.~~ 13-9.

[Change 8.5.8.3, p. 141]

#### **8.5.8.3 NORMAL to LIFE\_SAFETY\_ALARM Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15,~~ 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9.~~ 13-9.

[Change 8.5.8.4, p. 142]

#### **8.5.8.4 LIFE\_SAFETY\_ALARM to NORMAL Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15,~~ 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9.~~ 13-9.

[Change **8.5.8.5**, p. 142]

**8.5.8.5 LIFE\_SAFETY\_ALARM to OFFNORMAL Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9~~ 13-9.

[Change **8.5.8.6**, p. 142]

**8.5.8.6 OFFNORMAL to LIFE\_SAFETY\_ALARM Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9~~ 13-9.

[Change **8.5.8.7**, p. 142]

**8.5.8.7 Mode Transition Tests when Event State is Maintained**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9~~ 13-9.

[Change **8.5.8.8**, p. 143]

**8.5.8.8 NORMAL to OFFNORMAL Mode Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9~~ 13-9.

[Change **8.5.8.9**, p. 143]

**8.5.8.9 OFFNORMAL to NORMAL Mode Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9~~ 13-9.

[Change **8.5.8.10**, p. 143]

**8.5.8.10 NORMAL to LIFE\_SAFETY\_ALARM Mode Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9~~ 13-9.

[Change **8.5.8.11**, p. 143]

**8.5.8.11 LIFE\_SAFETY\_ALARM to NORMAL Mode Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9~~ 13-9.

[Change **8.5.8.12**, p. 144]

#### **8.4.8.12 LIFE\_SAFETY\_ALARM to OFFNORMAL Mode Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9~~ 13-9.

[Change **8.5.8.13**, p. 144]

#### **8.5.8.13 OFFNORMAL to LIFE\_SAFETY\_ALARM Mode Transition Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9~~ 13-9.

[Change **8.5.8.14**, p. 144]

#### **8.5.8.14 TO-FAULT Transition Tests**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: ~~12.11.7, 12.14, 12.15~~, 12.12.7, 12.15, 12.16, 13.3.8, ~~13.13~~ 13.9 and Figure ~~13.9~~ 13-9.

[Change **8.36**, p. 166]

#### **8.36 VT-Open Service Initiation Tests**

This clause defines the tests necessary to demonstrate support for initiating VT-Open service requests.

Dependencies: ReadProperty Service Execution tests, 9.18.

BACnet Reference Clauses: 17.2 and ~~12.10~~ 12.11.

[Change **8.37**, p. 167]

#### **8.37 VT-Close Service Initiation Tests**

This clause defines the tests necessary to demonstrate support for initiating VT-Close service requests. The IUT shall pass the test defined in 8.37.1. If the IUT supports more than one active VT session it shall also pass the tests defined in 8.37.2 and 8.37.3.

Dependencies: ReadProperty Service Execution tests, 9.18.

BACnet Reference Clauses: 17.3 and ~~12.10~~ 12.11.

[Change **9.28**, p. 262]

#### **9.28 ConfirmedTextMessage Service Execution Tests**

The purpose of this test group is to verify the correct execution of the ConfirmedTextMessage service request. BACnet does not define what is to happen when a ConfirmedTextMessage service request is received except that an acknowledgement is to be returned. It is likely that some other externally observable action will take place but this is vendor specific. These tests verify that a correct acknowledgment is returned and any other action that is defined by the vendor.



*BACnet Reference Clause: 16.5.*

[Change **13.2**, p. 425]

### **13.2 Time Master**

Purpose: To verify that an IUT can perform the function of a time master. To be a time master, a device must be capable of keeping time and of issuing TimeSynchronization service requests.

Dependencies: None.

BACnet Reference Clause: ~~12.10~~ 12.11.

## 135.1-2003c-2 Add new object types

### Rationale

Several object types have been added to ANSI/ASHRAE Standard 135 since the 2001 version. This revision adds object types that appear in the published 2004 version (without addenda). This addendum adds references to those objects only, not the tests for those objects.

### Addendum 135.1-2003c-2

[Change 4.5.4, p. 6]

[This clause is also changed in the proposed Addendum 135.1-2003b; the changes are additive and do not conflict]

### 4.5.4 Object Types Supported

...

The standard objects may be any of:

<del>Analog Input</del>	<del>Command</del>	<del>Multi-state Input</del>
<del>Analog Output</del>	<del>Device</del>	<del>Multi-state Output</del>
<del>Analog Value</del>	<del>Event Enrollment</del>	<del>Multi-state Value</del>
<del>Averaging</del>	<del>File</del>	<del>Notification Class</del>
<del>Binary Input</del>	<del>Group</del>	<del>Program</del>
<del>Binary Output</del>	<del>Life Safety Point</del>	<del>Schedule</del>
<del>Binary Value</del>	<del>Life Safety Zone</del>	<del>Trend Log</del>
<del>Calendar</del>	<del>Loop</del>	

<i>Accumulator</i>	<i>Command</i>	<i>Multi-state Output</i>
<i>Analog Input</i>	<i>Device</i>	<i>Multi-state Value</i>
<i>Analog Output</i>	<i>Event Enrollment</i>	<i>Notification Class</i>
<i>Analog Value</i>	<i>File</i>	<i>Program</i>
<i>Averaging</i>	<i>Group</i>	<i>Pulse Converter</i>
<i>Binary Input</i>	<i>Life Safety Point</i>	<i>Schedule</i>
<i>Binary Output</i>	<i>Life Safety Zone</i>	<i>Trend Log</i>
<i>Binary Value</i>	<i>Loop</i>	
<i>Calendar</i>	<i>Multi-state Input</i>	

[Insert new 4.5.10.1, p. 10, and renumber subsequent clauses]

[An identical addition is proposed by Addendum 135.1-2003b]

#### 4.5.10.1 Accumulator

```
{
  object-identifier: (accumulator, )
  object-name: ""
  object-type: accumulator
  present-value: 
  description: ""
  device-type: ""
  status-flags: 
  event-state: 
  reliability: 
  out-of-service: 
  scale: 
  units: 
  prescale: 
  max-pres-value: 
  value-change-time: 
```

```

value-before-change: □
value-set: □
logging-record: □
logging-object: □
pulse-rate: □
high-limit: □
low-limit: □
limit-monitoring-interval: □
notification-class: □
time-delay: □
limit-enable: {□,□}
event-enable: {□,□,□}
acked-transitions: {□,□,□}
notify-type: □
event-time-stamps: {□,□,□}
profile-name: "□"
}

```

[Insert new 4.5.10.23, p.19, and renumber subsequent clauses]  
[An identical addition is proposed by Addendum 135.1-2003b]

#### 4.5.10.23 Pulse Converter

```

{
object-identifier: (pulse-converter, □)
object-name: "□"
object-type: pulse-converter
description: "□"
present-value: □
input-reference: □
status-flags: □
event-state: □
reliability: □
out-of-service: □
units: □
scale-factor: □
adjust-value: □
count: □
update-time: □
count-change-time: □
count-before-change: □
cov-increment: □
cov-period: □
notification-class: □
time-delay: □
high-limit: □
low-limit: □
deadband: □
limit-enable: {□,□}
event-enable: {□,□,□}
acked-transitions: {□,□,□}
notify-type: □
event-time-stamps: {□,□,□}
profile-name: "□"
}

```

[Change ANNEX A, pp. 444-458]

## ANNEX A – EXAMPLE EPICS (INFORMATIVE)

(This annex is not part of the standard but is included for informative purposes.)

...

Standard Object Types Supported:

```
{
  Accumulator      Creatable  Deleteable
  Analog Input     Creatable  Deleteable
  ...
  Program          Creatable  Deleteable
  Pulse Converter  Creatable  Deleteable
  Schedule         Creatable  Deleteable
  Trend Log        Creatable  Deleteable
}
```

...

List of Objects in test device:

```
{
  {
    object-identifier: (accumulator, 1)
    object-name: "meter 1"
    object-type: ACCUMULATOR
    present-value: 125
    description: ""
    device-type: "electric pulse"
    status-flags: {FALSE,FALSE,FALSE,FALSE}
    event-state: NORMAL
    reliability: NO-FAULT-DETECTED
    out-of-service: FALSE
    scale: 0
    units: KILOWATT_HOURS
    prescale: (1,10000)
    max-pres-value: 9999
    value-change-time: {(Tuesday, 2-January-2007), 9:39:21.02}
    value-before-change: 0
    value-set: 67 W
    logging-record: {(Tuesday, 2-January-2007), 9:39:33.01}, 0, 27,NORMAL}
    logging-object: (trend-log, 3)
    pulse-rate: 3
    high-limit: 15
    low-limit: 0
    limit-monitoring-interval: 300
    notification-class: 1
    time-delay: 10
    limit-enable: {TRUE, TRUE}
    event-enable: {TRUE, FALSE, TRUE}
    acked-transitions: {TRUE, TRUE, TRUE}
    notify-type: ALARM
    event-time-stamps: {(Monday, 1-January-2007),18:50:21.02},
    {(*-*-*),*:*:*.}, {(Monday, 1-January-2007), 18:51:34.0}}
```

```

    profile-name: ""
  },
  ...
  {
    object-identifier: (device, 90)
    ...
    protocol-object-types-supported:
    {
      T, T, T, T, T, T,      -- AI, AO, AV, BI, BO, BV
      T, T, T, T, T, T      -- calendar, command, device, event enrollment, file, group
      T, T, T, T, T, F T,   -- loop, MSI, MSO, notification class, program, schedule
      T, T, T, T, T,       -- averaging, multi-state-value, trend-log, life-safety-point, life-safety-zone
      T, T                  -- accumulator, pulse-converter
    }
    object-list:
    {
      (accumulator, 1),
      ...
      (analog-value, 1),
      (averaging, 1),
      (binary-input, 1),
      ...
      (group, 1),
      (life-safety-point, 1),
      (life-safety-zone, 1),
      (loop, 1),
      ...
      (program, 1),
      (pulse-converter, 1),
      (schedule, 1),
      (trend-log, 1),
      ...
    }
  },
  ...
  {
    object-identifier: (pulse-converter, 1)
    object-name: "Meter 5"
    object-type: PULSE_CONVERTER
    description: ""
    present-value: 125.0
    input-reference: {(accumulator, 1), present-value}
    status-flags: {FALSE, FALSE, FALSE, FALSE}
    event-state: NORMAL
    reliability: NO-FAULT-DETECTED
    out-of-service: FALSE
    units: LITERS_PER_HOUR
    scale-factor: 0.5
    adjust-value: 500.0
    count: 250
    update-time: {(Thursday, 4-January-2007), 9:39:21.02}
    count-change-time: {(Thursday, 4-January-2007), 9:39:41.52}
    count-before-change: 523
    cov-increment: 10.0
    cov-period: 3600
  }
}

```

```

notification-class: 1
time-delay: 0
high-limit: 1000.0
low-limit: 0.0
deadband: 0.0
limit-enable: {FALSE, TRUE}
event-enable: {TRUE, FALSE, TRUE}
acked-transitions: {TRUE, TRUE, TRUE}
notify-type: ALARM
event-time-stamps: {{(Monday, 1-January-2007),18:50:21.02},
  {(*-*-*)*:*:*.*},{(Monday, 1-January-2007), 18:51:34.0}}
profile-name: ""
},
...
{
  object-identifier: (trend-log, 1)
  object-name: "Room 3 Log"
  object-type: TREND_LOG
  description: "Room 3 Temperature"
  log-enable: TRUE
  log-deviceobjectproperty: {(device, 100), (analog input, 3), present-value}
  log-interval: 6000
  stop-when-full: FALSE
  buffer-size: 250
  log-buffer: ()
  record-count: 250
  total-record-count: 131040
  notification-threshold: 83
  records-since-notification: 30
  last-notify-record: 131010
  event-state: NORMAL
  notification-class: 1
  event-enable: {FALSE, TRUE, TRUE}
  acked-transitions: {TRUE, TRUE, TRUE}
  notify-type: EVENT
  event-time-stamps: {{(Monday, 1-January-2007),18:50:21.02},
    {(*-*-*)*:*:*.*},{(Monday, 1-January-2007), 18:51:34.0}}
  profile-name: ""
}
}
End of BACnet Protocol Implementation Conformance Statement

```

## 135.1-2003c-3 Omit certain tests based on Protocol\_Revision

### Rationale

Several test were made invalid by changes to 135-2004. This revision checks the Protocol\_Revision property to determine test validity.

### Addendum 135.1-2003c-3

[Change 7.3.2.22.3.2, p. 76]

#### 7.3.2.22.3.2 Calendar Entry Date Test

...

Configuration Requirements: The IUT shall be configured to contain a Schedule object with an Exception\_Schedule containing a BACnetCalendarEntry with a specific date. The criteria for the dates used in the test are given in Table 7-4. The local date and time shall be set such that the Present\_Value property has a value other than  $V_1$ . *This test shall not be performed if the Protocol\_Revision property is present in the Device object and has a value of 4 or greater.*

[Change 7.3.2.22.3.3, p. 76]

#### 7.3.2.22.3.3 Calendar Entry DateRange Test

...

Configuration Requirements: The IUT shall be configured to contain a Schedule object with an Exception\_Schedule containing a BACnetCalendarEntry with a date range. The criteria for the dates used in the test are given in Table 7-5. The local date and time shall be set such that the Present\_Value property has a value other than  $V_1$ . *This test shall not be performed if the Protocol\_Revision property is present in the Device object and has a value of 4 or greater.*

[Change 7.3.2.22.3.4, p. 77]

#### 7.3.2.22.3.4 Calendar Entry WeekNDay Month Test

...

Configuration Requirements: The IUT shall be configured to contain a Schedule object with an Exception\_Schedule containing a BACnetCalendarEntry with a WeekNDay entry specifying a month. The criteria for the dates used in the test are given in Table 7-6. The local date and time shall be set such that the Present\_Value property has a value other than  $V_1$ . *This test shall not be performed if the Protocol\_Revision property is present in the Device object and has a value of 4 or greater.*

[Change 7.3.2.22.3.5, p. 78]

#### 7.3.2.22.3.5 Calendar Entry WeekNDay Week Of Month Test

...

Configuration Requirements: The IUT shall be configured to contain a Schedule object with an Exception\_Schedule containing a BACnetCalendarEntry with a WeekNDay entry specifying a week of the month. The criteria for the dates used in the test are given in Table 7-7. The local date and time shall be set such that the Present\_Value property has a value other than  $V_1$ . *This test shall not be performed if the Protocol\_Revision property is present in the Device object and has a value of 4 or greater.*

[Change 7.3.2.22.3.6, p. 78]

#### **7.3.2.22.3.6 Calendar Entry WeekNDay Last Week Of Month Test**

...

Configuration Requirements: The IUT shall be configured to contain a Schedule object with an Exception\_Schedule containing a BACnetCalendarEntry with a WeekNDay entry specifying the last week of the month. The criteria for the dates used in the test are given in Table 7-8. The local date and time shall be set such that the Present\_Value property has a value other than  $V_1$ . *This test shall not be performed if the Protocol\_Revision property is present in the Device object and has a value of 4 or greater.*

[Change 7.3.2.22.3.7, p. 79]

#### **7.3.2.22.3.7 Calendar Entry WeekNDay Day Of Week Test**

...

Configuration Requirements: The IUT shall be configured to contain a Schedule object with an Exception\_Schedule containing a BACnetCalendarEntry with a WeekNDay entry specifying the day of the week. The criteria for the dates used in the test are given in Table 7-9. The local date and time shall be set such that the Present\_Value property has a value other than  $V_1$ . *This test shall not be performed if the Protocol\_Revision property is present in the Device object and has a value of 4 or greater.*

[Change 7.3.2.22.5, p. 81]

#### **7.3.2.22.5 Exception\_Schedule Restoration Test**

...

Configuration Requirements: The IUT shall be configured with a Schedule object that contains an Exception\_Schedule that has more than one scheduled write operation for a particular day. None of the write operations shall be scheduled for time 00:00 and there shall be no higher priority coincident BACnetSpecialEvents. In the test description  $D_1$  represents a time between 00:00 on the day the Exception\_Schedule is active and the time of the first schedule write operation in the BACnetSpecialEvent.  $V_{last}$  represents the value that is scheduled to be written in the last BACnetTimeValue pair for the day. *This test shall not be performed if the Protocol\_Revision property is present in the Device object and has a value of 4 or greater.*

[Change 7.3.2.22.6, p. 82]

#### **7.3.2.22.6 Weekly\_Schedule Restoration Test**

...

Configuration Requirements: The IUT shall be configured with a Schedule object that contains a Weekly\_Schedule that has more than one scheduled write operation for a particular day. None of the write operations shall be scheduled for time 00:00 and there shall be no higher priority coincident BACnetSpecialEvents. In the test description  $D_1$  represents a time between 00:00 and the time of the first scheduled write operation in the BACnetDailySchedule.  $V_{last}$  represents the value that is scheduled to be written in the last BACnetTimeValue pair for the day. *This test shall not be performed if the Protocol\_Revision property is present in the Device object and has a value of 4 or greater.*



[Change 7.3.2.23.11, p. 92]

#### **7.3.2.23.11 Notification Time Tests**

...

Purpose: To verify that the Previous\_Notify\_Time and Current\_Notify\_Time properties reflect the values sent in the most recent notification. *This test shall be performed only if the Protocol\_Revision property is present in the Device object and has a value of 1 or 2.*

## 135.1-2003c-4 Exception schedule priority requirements

### Rationale

Proper testing requires that there is no interference from higher priority exception schedules.

### Addendum 135.1-2003c-4

[Change 7.3.2.22.3.9, p. 80]

#### 7.3.2.22.3.9 List of BACnetTimeValue Test

...

Configuration Requirements: The IUT shall be configured with a Schedule object containing a ~~BACnetSpecialEvents~~ *BACnetSpecialEvent* with two or more BACnetTimeValue ~~entries~~ *entries and no BACnetSpecialEvents with a higher priority*. Each BACnetTimeValue entry shall have a distinguishable value.

[Change 7.3.2.22.7, p. 83]

#### 7.3.2.22.7 List\_Of\_Object\_Property\_Reference Internal Test

...

Configuration Requirements: The IUT is configured with a Schedule object containing a List\_Of\_Object\_Property\_References property that references, if possible, at least one property in another object within the IUT. The Schedule object is configured with either a Weekly\_Schedule or an active Exception\_Schedule, during a period where Effective\_Period is active, with at least two consecutive entries with distinguishable values in the List of ~~BACnetTimeValues~~ *BACnetTimeValues, and with no Exception\_Schedules at a higher priority*.  $D_1$  represents the date and time of the first of these two BACnetTimeValues, with corresponding value  $V_1$ , while  $D_2$  and  $V_2$  (a value distinguishable from  $V_1$ ) represent the second BACnetTimeValue. A time  $D_i$  is defined to occur between  $D_1$  and  $D_2$ .

[Change 7.3.2.22.8, p. 83]

#### 7.3.2.22.8 List\_Of\_Object\_Property\_Reference External Test

...

Configuration Requirements: The TD is configured to indicate that it supports the WriteProperty-Request service but not WritePropertyMultiple-Request. The IUT is configured with a Schedule object containing a List\_Of\_Object\_Property\_References property that references a property of an object contained in the TD. The Schedule object is configured with either a Weekly\_Schedule or an active Exception\_Schedule, during a period where Effective\_Period is active, with at least two consecutive entries with distinguishable values in the List of ~~BACnetTimeValues~~ *BACnetTimeValues, and with no Exception\_Schedules at a higher priority*.  $D_1$  represents the date and time of the first of these two BACnetTimeValues, with corresponding value  $V_1$ , while  $D_2$  and  $V_2$  (a value distinguishable from  $V_1$ ) represent the second BACnetTimeValue. A time  $D_i$  is defined to occur between  $D_1$  and  $D_2$ .

## 135.1-2003c-5 Minor corrections

### Rationale

Several small changes are required to correct or update the standard to match standard 135-2004.

### Addendum 135.1-2003c-5

[Change 4.1, pp. 1-2]

...

**Table 4-1.** Character Set Codes

<i>code</i>	<i>character set</i>
0	ANSI <del>3.4</del> x3.4
1	Microsoft DBCS
2	JIS C 6226
3	ISO 10646 (UCS-4)
4	ISO 10646 (UCS-2)
5	ISO 8859-1

An octet stream format can be recognized by examining the first eight octets of the EPICS file. Using ANSI ~~3.4~~ X3.4 encoding as an example these eight octets will contain: X'50' X'49' X'43' X'53' X'20' X'30' X'0D' X'0A'. This represents the text "PICS 0" followed by carriage return and linefeed.

[Change 4.5.5, p. 7]

#### 4.5.5 Data Link Layer Options

...

```
BACnet/IP, 'DIX' Ethernet.┘  
BACnet/IP, PPP.┘ Other:  ┘  
Other.┘  
}┘
```

[Change 7.3.2.23.9, p. 90]

#### 7.3.2.23.9 Total\_Record\_Count Test

...

Purpose: To verify that the Total\_Record\_Count property increments for each record added to the Log\_Buffer, even after Buffer\_Size records have been added. (Note: it is not reasonable to test for the requirement of BACnet Clause 12.23.16 that the value wrap from  $2^{32}-1$  to 0; even if a record was collected every 100<sup>th</sup> of a second it could take more than 497 days to complete the test.)

[Change ANNEX A, pp. 444-458]

## ANNEX A – EXAMPLE EPICS (INFORMATIVE)

(This annex is not part of the standard but is included for informative purposes.)

...

Data Link Layer Option:

```
{
  ...
  BACnet/IP, 'DIX' Ethernet
  BACnet/IP, PPP Other: _____
  LonTalk
  Other
}
```

...

Character Sets Supported:

```
{
  ANSI X3.4
  IBM/Microsoft DBCS
  ISO 8859-1
  JIS C 6226
  ISO 10646 (UCS-4) (UCS-4)
  ISO 10646 (UCS-2) (UCS-2)
}
```

...

Special Functionality:

```
{
  Maximum APDU size in octets: 1470 1476
  ...
}
```

...

List of Objects in test device:

```
{
  ...
  {
    object-identifier: (analog-input, 1)
    ...
    out-of-service: F FALSE
    ...
  },
  ...
  {
    object-identifier: (device, 90)
    ...
    protocol-revision: 2 4
    protocol-services-supported:
    {
      T, T, T, T, T, T, -- Alarm and event
      T, T, -- File
      T, T, T, T, T, T, -- Create, Delete, Read
      T, T, T, -- ReadMultiple, Write, WriteMultiple
      T, T, T, T, -- PrivateXfer, Re-init
      T, T, T, -- VT Open, Data, Close
      T, T, -- security
      T, T, T, T, T, -- I-Am, I-Have, uPrivateXfer
      T, T, T, T -- TimeSync, Who-Has, Who-Is
    }
  }
}
```

```

    T, T, T, T,      -- ReadRange, UTCTimeSync, LifeSafetyOperation, SubscribeCOVProperty
    T                -- GetEventInfo
  }
  ...
},
...
{
  object-identifier: (schedule, 2)
  ...
  exception-schedule: { {(23-November-1995),({0:00,INACTIVE}),10},
  schedule-default: INACTIVE
  list-of-object-property-references: ((binary-output, 9),present-value)
  priority-for-writing: 15
  reliability: NO_FAULT_DETECTED
  out-of-service: FALSE
  profile-name: ""
},
...
}
....

```

[Add a new **History of Revisions**, p.459, with the following entries]

(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	N/A	<p><b>Addendum a to ANSI/ASHRAE 135.1-2003</b>            Approved by the ASHRAE Standards Committee January 21, 2006; by the ASHRAE Board of Directors January 26, 2006; and by the American National Standards Institute January 27, 2006.</p> <ol style="list-style-type: none"> <li>1. Add Partial Day Scheduling to Schedule object.</li> <li>2. Enable reporting of proprietary events by the Event Enrollment object.</li> <li>3. Allow detailed error reporting when all ReadPropertyMultiple accesses fail.</li> <li>4. Remove the Recipient property from the Event Enrollment object.</li> <li>5. MS/TP slave proxy tests.</li> <li>6. Add a new silenced mode to the DeviceCommunicationControl service.</li> <li>7. Addition of tests for Data Sharing BIBBs.</li> <li>8. Specify the behavior of a BACnetARRAY when its size is changed.</li> <li>9. Clarifying the behavior of a BACnet router when it receives an unknown network message type.</li> <li>10. Testing unsupported service request execution.</li> <li>11. Reading entire arrays.</li> <li>12. Update negative tests.</li> </ol>
1	N/A	<p><b>Addendum c to ANSI/ASHRAE 135.1-2003</b>            Approved by the ASHRAE Standards Committee June 23, 2007; by the ASHRAE Board of Directors June 27, 2007; and by the American National Standards Institute June 28, 2007.</p> <ol style="list-style-type: none"> <li>1. Update references to refer to 135-2004.</li> <li>2. Add new object types from 135-2004.</li> <li>3. Omit certain tests based on Protocol_Revision.</li> <li>4. Exception schedule priority requirements.</li> <li>5. Minor corrections.</li> </ol>

**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.