ANSI/ASHRAE Addendum i to
ANSI/ASHRAE Standard 135.1-2009

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.

ISSN 1041-2336

Approved American National Standard
ANSI
www.ansi.org

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

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.

[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*i* 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.

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 ~~strikethrough~~. Where entirely new subclauses are added, plain type is used throughout.

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-1. Improve Schedule Object Restoration Tests.

Rationale

ASHRAE 135.1 tests 7.3.2.23.5 and 7.3.2.23.6 examine whether a Schedule object will write the correct value from its Exception_Schedule and Weekly_Schedule properties when the device containing that schedule is reinitialized via the ReinitializeDevice service. These tests specify that the 'COLDSTART' argument shall be used when the device is reinitialized, but the BACnet standard does not define the meaning of 'COLDSTART'. Instead, these tests should specify the 'WARMSTART' argument.

[Change **Clause 7.3.2.23.5** Exception_Schedule Restoration Test, p. 91 and
Clause 7.3.2.23.6 Weekly_Schedule Restoration Test, p. 91]

…
4. IF (ReinitializeDevice execution is supported) THEN
TRANSMIT ReinitializeDevice-Request,
'Reinitialized State of Device' = ~~COLDSTART~~ *WARMSTART*,
'Password' = (any valid password)
RECEIVE BACnet-Simple-ACK-PDU
ELSE
MAKE (the IUT reinitialize)
5. CHECK (Did the IUT perform a ~~COLDSTART~~ *WARMSTART* reboot?)
…

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-2. Add Text to Check Range of the Present_Value of Multi-state Objects.

Rationale

This new test verifies that the Present_Value property of a Multi-state Input, Multi-state Output or Multi-state Value object remains within its valid range of values.

[Add new **Clause 7.3.1.X**, p. 49]

7.3.1.X Number_Of_States Range Test

Dependencies: ReadProperty Service Execution Tests, 7.1; WriteProperty Service Execution Tests 7.2.2

BACnet Reference Clauses: 12.18.11, 12.19.4, 12.19.11, 12.20.4, 12.20.10

Purpose: This test case verifies that the Present_Value property of a Multi-state Input, Multi-state Output or Multi-state Value object is limited to the range 1 through the value of its Number_Of_States property.

Test Concept: The Number_Of_States property is first verified to be a non-zero value. An attempt is then made to modify the Present_Value property of the referenced object to contain a value greater than the value of the Number_Of_States property. An attempt is then made to set the Present_Value property to 0. The test shall verify a correct error response for each case.

Configuration Requirements: The IUT shall be configured to contain a Multi-state Input, Multi-state Output, or Multi-state Value object, Object1, which contains a writable Present_Value property. Object1 shall be configured such that the Present_Value property is writable before executing the test. If the Present_Value property cannot be made writable, then this test shall be skipped.

Test Steps:

1. READ COUNT = Number_Of_States
2. VERIFY Number_Of_States = (a non-zero value)
3. TRANSMIT WriteProperty-Request,
   'Object Identifier' = Object1,
   'Property Identifier' = Present_Value,
   'Property Value' = (any value larger than COUNT)
4. RECEIVE BACnet-Error-PDU,
   Error Class = PROPERTY,
   Error Code = VALUE_OUT_OF_RANGE
5. VERIFY (Object1), P1 = (a value between 1 and COUNT, inclusive)
6. TRANSMIT WriteProperty-Request,
   'Object Identifier' = Object1,
   'Property Identifier' = Present_Value,
   'Property Value' = 0
7. RECEIVE BACnet-Error-PDU,
   Error Class = PROPERTY,
   Error Code = VALUE_OUT_OF_RANGE
8. VERIFY (Object1), P1 = (a value between 1 and COUNT, inclusive)

Note To Tester: When testing an object that is commandable, any priority may be selected.

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-3. Add Test for SubscribeCOV Service Execution Without a Lifetime Parameter.

Rationale

Add a test to ensure that the IUT correctly processes SubscribeCOV requests which do not include a Lifetime parameter.

[Add new **Clause 8.2.X1,** p. 116]

8.2.x1 Missing Lifetime Test

Purpose: This test case verifies the special case of the SubscribeCOV where a missing Lifetime parameter shall imply an indefinite Lifetime subscription.

Test Concept: A subscription for COV notification is established with the IUT. The subscribe message shall omit the Lifetime parameter. The COV notification is received from the IUT and the 'Time Remaining' value is verified to be 0.

Test Steps:

1. TRANSMIT SubscribeCOV-Request,
   'Subscriber Process Identifier' = (any value > 0 chosen by the TD),
   'Monitored Object Identifier' = X,
   'Issue Confirmed Notifications' = TRUE
2. RECEIVE BACnet-SimpleACK-PDU,
3. RECEIVE ConfirmedCOVNotification-Request,
   'Subscriber Process Identifier' = (the same value used in step 1),
   'Initiating Device Identifier' = IUT,
   'Monitored Object Identifier' = X,
   'Time Remaining' = 0,
   'List of Values' = (the initial Present_Value and initial Status_Flags)
4. TRANSMIT BACnet-SimpleACK-PDU

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-4. Update Test for Processing of ReadProperty Service Responses.

Rationale

Update the ReadProperty service initiation tests to provide a response to the IUT and verify that the IUT properly processes the responses. Add new ReadProperty service initiation tests to cover cases currently not tested.

[Change **Clause 8.18.1,** p. 165]

8.18.1 Reading Non-Array Properties

Purpose: To verify that the IUT can initiate a ReadProperty service request that does not contain the 'Property Array Index' ~~parameter.~~ *parameter and can correctly process the response.*

Test Steps:

1. RECEIVE ReadProperty-Request,
   'Object Identifier' = (any object),
   'Property Identifier' = (any valid non-array property of the specified object)
2. *TRANSMIT BACnet-ComplexACK-PDU,*
   'Object Identifier' = (object identifier from step 1),
   'Property Identifier' = (property identifier from step 1),
   'Property Value' = (any valid value for the property)
3. *CHECK (that the IUT exhibits the vendor defined results)*

[Change **Clause 8.18.2**, p. 165]

8.18.2 Reading an Array Element

Purpose: To verify that the IUT can initiate a ReadProperty service request that references a specific element of an array ~~property.~~*property and can correctly process the response.*

Test Steps:

1. RECEIVE ReadProperty-Request,
   'Object Identifier' = (any object),
   'Property Identifier' = (any valid array property of the specified object),
   'Array Index' = (any valid array index for the specified property)
2. *TRANSMIT BACnet-ComplexACK-PDU,*
   'Object Identifier' = (object identifier from step 1),
   'Property Identifier' = (property identifier from step 1),
   Array Index' = (array index from step 1),
   'Property Value' = (any valid value for the property)
3. *CHECK (that the IUT exhibits the vendor defined results)*

[Add new **Clause 8.18.X1,** p. 165]

8.18.X1 Reading Whole Array Properties

Purpose: To verify that the IUT can initiate a ReadProperty service request that does not contain the 'Property Array Index' parameter for an array property and can correctly process the response.

Test Steps:

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

1. RECEIVE ReadProperty-Request,
   'Object Identifier' = (any object),
   'Property Identifier' = (any valid array property of the specified object)
2. TRANSMIT BACnet-ComplexACK-PDU,
   'Object Identifier' = (object identifier from step 1),
   'Property Identifier' = (property identifier from step 1),
   'Property Value' = (any valid array of values for the property)
3. CHECK (that the IUT exhibits the vendor defined results)

[Add new **Clause 8.18.X2,** p. 165]

8.18.X2 Reading an Array Size

Purpose: To verify that the IUT can initiate a ReadProperty service request for the size of an array property and can correctly process the response.

Test Steps:

1. RECEIVE ReadProperty-Request,
   'Object Identifier' = (any object),
   'Property Identifier' = (any valid array property of the specified object),
   'Array Index' = 0
2. TRANSMIT BACnet-ComplexACK-PDU,
   'Object Identifier' = (object identifier from step 1),
   'Property Identifier' = (property identifier from step 1),
   Array Index' = 0,
   'Property Value' = (any valid size for the array)
3. CHECK (that the IUT exhibits the vendor defined results)

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-5. Add Tests for Processing of GetEventInformation Service Responses.

Rationale

Update the GetEventInformation service initiation tests to provide a response to the IUT and verify that the IUT properly processes the responses.

[Change **Clause 8.8.1,** p. 158]

8.8.1 Without Chaining

Purpose: To verify that the IUT can initiate a simple GetEventInformation service request.

Test Steps:

1. RECEIVE GetEventInformation-Request,
   'Last Received Object Identifier' = (none)
2. *TRANSMIT BACnet-ComplexACK-PDU,*
   'List of Event Summaries' = (any valid list)
   'More Events' = FALSE
3. *CHECK(that the IUT exhibits the vendor defined results)*

[Change **Clause 8.8.2,** p. 158]

8.8.2 With Chaining

Purpose: To verify that the IUT can respond to the "more events" flag by initiating a GetEventInformation service request with the chaining parameter "last received object identifier" present.

Test Steps:

1. RECEIVE GetEventInformation-Request,
   'Last Received Object Identifier' = (none)
2. TRANSMIT BACnet-ComplexACK-PDU,
   'List of Event Summaries' = (a non-empty list)
   'More Events' = TRUE
3. BEFORE the tester's patience is exceeded
   RECEIVE GetEventInformation-Request,
   'Last Received Object Identifier' = (last object identifier sent in step 2)
4. *TRANSMIT BACnet-ComplexACK-PDU,*
   'List of Event Summaries' = (a non-empty list)
   'More Events' = FALSE
5. *CHECK(that the IUT exhibits the vendor defined results)*

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-6. Add Tests for Fallback from ReadPropertyMultiple to ReadProperty.

Rationale

Add tests to verify that IUTs which initiate ReadPropertyMultiple are able to use ReadProperty instead when the other does not support ReadPropertyMultiple.

[Add new **Clause 8.20.Y1,** p. 167]

8.20.Y1 Cases In Which ReadProperty Shall Be Used After ReadPropertyMultiple Fails

The tests defined in this clause are used to verify that an IUT which intiates ReadPropertyMultiple is able to obtain external property values via the ReadProperty service when interoperating with a device that does not support the ReadPropertyMultiple service.

8.20.Y1.1 The IUT Determines the TD does not Support the ReadPropertyMultiple Service

Purpose: Verifies the IUT's ability to automatically change its service choice from ReadPropertyMultiple to ReadProperty when the IUT determines the TD does not support the ReadPropertyMultiple service.

Test Concept: The IUT is configured in a manner that would normally cause it to access one or more properties in the TD via the ReadPropertyMultiple service. Prior to sending a ReadPropertyMultiple request, however, the IUT determines that the TD does not support the ReadPropertyMultiple service. The IUT instead attempts to access the TD's property values via the ReadProperty service (it is assumed that the IUT will make this determination by reading the TD's Protocol_Services_Supported property, but this test specifically does not attempt to verify this behavior).

Configuration Requirements: The TD is configured so that it does not support the ReadPropertyMultiple service. The IUT is configured such that it is capable of accessing one or more properties of a single object in the TD via the ReadProperty and ReadPropertyMultiple services. If the IUT cannot be configured in this way, then this test shall be omitted.

Test Steps:

1. MAKE (a condition in the IUT that would normally cause it to send a ReadPropertyMultiple request to the TD to access one or more properties values of a single object)
2. WAIT (a time interval specified by the vendor as sufficient for the IUT to determine that the TD does not support the ReadPropertyMultiple service)
3. REPEAT X = (the properties that the IUT is to read) DO {
   RECEIVE ReadProperty-Request,
   'Object Identifier' = (object identifier referenced by X),
   'Property Identifier' = (property identifier referenced by X)
   TRANSMIT ReadProperty-Ack,
   'Object Identifier' = (object identifier referenced by X),
   'Property Identifier' = (property identifier referenced by X),
   'Property Value' = (any valid value)
   }

8.20.Y1.2 Fallback to ReadProperty on Reject - UNRECOGNIZED_SERVICE Response

BACnet Reference Clauses: 15.5 and 15.7

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

Purpose: Verifies the IUT's ability to automatically change its service choice from ReadPropertyMultiple to ReadProperty when the TD returns a Reject-PDU and a Reject Reason of UNRECOGNIZED_SERVICE.

Test Concept: The IUT is configured to send the TD a ReadPropertyMultiple request to access one or more properties of a single object. The TD responds with a Reject-PDU and a Reject Reason of UNRECOGNIZED_SERVICE. With no additional configuration, the IUT sends one or more ReadProperty requests to the TD, where each ReadProperty request specifies an individual property from the original ReadPropertyMultiple request.

Configuration Requirements: The TD is configured so that it does not support the ReadPropertyMultiple service. The IUT is configured such that it attempts to acquire values from the TD using the ReadPropertyMultiple service without first interrogating the TD's Protocol_Services_Supported property. If the IUT cannot be configured in this way then this test shall be omitted.

Test Steps:

1. RECEIVE ReadPropertyMultiple-Request,
   'Object Identifier' = (object identifier of the specified object),
   'List of Property References' = (one or more properties of the specified object)
2. TRANSMIT BACnet-Reject-PDU,
   'Reject Reason' = UNRECOGNIZED_SERVICE
3. REPEAT X = (the properties from Step 1) DO {
   RECEIVE ReadProperty-Request,
   'Object Identifier' = (object identifier referenced by X),
   'Property Identifier' = (property identifier referenced by X)
   TRANSMIT ReadProperty-Ack,
   'Object Identifier' = (object identifier referenced by X),
   'Property Identifier' = (property identifier referenced by X),
   'Property Value' = (any valid value)
   }

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-7. Allow Priorities in WriteProperty and WritePropertyMultiple Tests.

Rationale

These changes allow an IUT to include a priority when generating WriteProperty and WritePropertyMultiple service requests.

This note is to be applied to all tests in the standard and all addenda where the IUT is expected to generate a WriteProperty or WritePropertyMultiple request that is not already expecting a Priority parameter to be present.

[Append to **Clause 7.3.2.9.3,** p. 59, **Clause 7.3.2.23.8,** p. 93, **Clause 8.22.1,** p. 168, **Clause 8.22.2,** p. 169 and to any clause in any addendum that expects the IUT to generate an WriteProperty and does not expect a Priority parameter.]

Note to Tester: Any WriteProperty request generated by the IUT may have a Priority parameter. If included, it shall be in the range 1-16, excluding 6.

[Append to **Clause 8.23.1,** p. 169, **Clause 8.23.2,** p. 169, **Clause 8.23.3,** p. 170, **Clause 8.23.4,** p. 170, and **Clause** 8.23.5, p. 170 and to any clause in any addendum that expects the IUT to generate an WritePropertyMultiple and does not expect a Priority parameter.]

Note to Tester: Any WritePropertyMultiple request generated by the IUT may have a Priority parameter for any of the properties written. If included, the Priority parameter shall be in the range 1-16, excluding 6 .

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-8. Add Test for Writing Array Size.

Rationale

Add in a test that verifies that the IUT can modify the size of an array.

[Add new **Clause 8.22.X1**, p. 169]

8.22.X1 Writing An Array Size

Purpose: To verify that the IUT can initiate WriteProperty service requests that modify the size of an array.

Test Steps:

1. RECEIVE WriteProperty-Request,
   'Object Identifier' = (any valid object identifier),
   'Property Identifier' = (any valid array property of the specified object),
   'Array Index' = 0
   'Property Value' = (any unsigned value)

Note to Tester: Any WriteProperty request generated by the IUT may have a Priority parameter. If included, it shall be in the range 1-16, excluding 6.

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-9. Clarify Test for Writing with a Value that is Out of Range.

Rationale

Clarify the requirements for a value that is out of range and ensure that the tests are executed against writable properties.

[Change **Clause 9.22.2.4**, p. 269]

9.22.2.4 Writing with a Property Value that is Out of Range

Purpose: To verify that the IUT can execute WriteProperty service requests when an attempt is made to write a value that is outside of the supported range.

Test Concept: The TD attempts to write to a property using a value that is outside of the supported range. *If the IUT does not contain any writable properties that have restricted ranges, then this test shall be skipped.*

Test Steps:

1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS),
2. TRANSMIT WriteProperty-Request,
   'Object Identifier' = (Object1, any object with writable properties),
   'Property Identifier' = (P1, any *writable* property with a restricted range of values),
   'Property Value' = (any value that is outside the supported range)
3. IF (Protocol_Revision is present and Protocol_Revision ≥ 4) THEN
   RECEIVE (BACnet-Error PDU,
   Error Class = PROPERTY,
   Error Code = VALUE_OUT_OF_RANGE
   ELSE
   RECEIVE (BACnet-Error-PDU,
   Error Class = PROPERTY,
   Error Code = VALUE_OUT_OF_RANGE) |
   (BACnet-Reject-PDU,
   Reject Reason = PARAMETER_OUT_OF_RANGE)
4. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)

Notes to Tester: The value used in step 2 shall be of the correct datatype. For bit string types, the bit count shall be correct for Date and Time values, the value shall be within the range defined by the standard for the datatype and for constructed values, the constructed value shall match the structure defined by the ASN.1, and all field values shall be within the ranges defined by the standard for those field values.

[Add new **Clause 9.23.1.X**, p. 274]

9.23.1.X Writing to Properties Based on Data Type

Purpose: This test case verifies that the IUT can execute WritePropertyMultiple service requests to any data type supported by the IUT.

Test Concept: For the specified data type, the TD shall select an object in the IUT that contains a writable property of that data type. This property is designated P1.

Configuration Requirements: Configure the IUT with at least one writable property of the data type that can be used for this test.

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

Test Steps:

1. READ V = Object1, P1
2. TRANSMIT WritePropertyMultiple-Request,
   'Object Identifier' = Object1,
   'Property Identifier' = P1,
   'Property Value' = (any valid value defined for this property subject to the restrictions specified in the EPICS as defined in Clause 4.4.2, except the value, V, read in step 1)
3. RECEIVE Simple-ACK-PDU
4. VERIFY (Object1), P1 = (the value used in step 2)

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-10. Update Test for Writing with an Invalid Datatype.

Rationale

Update the invalid datatype tests to take into account the committee's response to IC-2004-28 and changes made in Protocol_Revision 7.

The tests are also updated to not rely on the EPICS value and to use the new READ keyword.

[Change **Clause**, **9.22.2.3**, p. 268]

9.22.2.3 Writing with a Property Value Having the Wrong Datatype

Purpose: To verify that the IUT correctly responds to an attempt to write a property value that has an invalid datatype.

Test Concept: The TD shall select an object in the IUT that contains a writable property designated P1. An attempt will be made to write to this property using ~~an invalid datatype~~ *a datatype that the IUT supports but which is invalid for the property*. If no object supports writable properties, then this test shall be omitted.

Test Steps:

~~1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)~~
1. READ V = (Object1), P1
2. TRANSMIT WriteProperty-Request,
'Object Identifier' = Object1,
'Property Identifier' = P1,
'Property Value' = (any value with an invalid datatype)
3. ~~IF (Protocol_Revision is present and Protocol_Revision ≥ 4) THEN~~
~~RECEIVE BACnet-Error PDU,~~
~~Error Class = PROPERTY,~~
~~Error Code = INVALID_DATATYPE~~
~~ELSE~~
RECEIVE (BACnet-Error PDU,
Error Class = PROPERTY,
Error Code = INVALID_DATATYPE) |
(BACnet-Reject-PDU
Reject Reason = INVALID_PARAMETER_DATATYPE) |
(BACnet-Reject-PDU
Reject Reason = INVALID_TAG)
4. VERIFY (Object1), P1 = *V*~~(the value defined for this property in the EPICS)~~

[Change **Clause 9.23.2.6**, p. 277]

9.23.2.6 Writing with a Property Value Having the Wrong Datatype

Purpose: This test case verifies that the IUT correctly responds to an attempt to write a property value that has an invalid datatype.~~ This test shall only be performed if Protocol_Revision is present and has a value greater than or equal to 4.~~

Test Concept: The TD shall select an object, designated Object1, in the IUT that contains a writable property designated P1. An attempt will be made to write to this property using ~~an invalid datatype~~ *a datatype that the IUT supports but which is invalid for the property*. If no object supports writable properties, then this test shall be omitted.

Test Steps:

~~1. VERIFY (Object1), P1 = (the value defined for this property in the EPICS)~~

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

1. *READ V = (Object1), P1*
2. TRANSMIT WritePropertyMultiple-Request,
   'Object Identifier' = Object1,
   'Property Identifier' = P1,
   'Property Value' = (any value with an invalid datatype)
3. RECEIVE
   WritePropertyMultiple-Error,
   'Error Class' = PROPERTY,
   'Error Code' = INVALID_DATATYPE
   'Object Identifier' = Object1
   'Property Identifier' = P1
   | *(BACnet-Reject-PDU*
   'Reject Reason' = INVALID_PARAMETER_DATATYPE)
   | *(BACnet-Reject-PDU*
   'Reject Reason' = INVALID_TAG)
4. VERIFY (Object1), P1 = *V*~~(the value defined for this property in the EPICS)~~

[Change **Clause 9.14.2.3**, p 245]
[Step number changes are not change marked for clarity.]

9.14.2.3 An AddListElement Failure Part Way Through a List

Purpose: To verify the ability of the IUT to respond to an AddListElement service request to add multiple elements to a list where one of the elements cannot be added. Upon failure, the AddListElement service should leave the list unchanged.

Test Steps:

~~1. TRANSMIT ReadProperty-Request,~~
~~'Object Identifier' = L,~~
~~'Property Identifier' = ListProp~~
~~2. RECEIVE ReadProperty-ACK,~~
~~'Object Identifier' = L,~~
~~'Property Identifier' = ListProp,~~
~~'Property Value' = (any valid value referred to as "InitialList" below)~~

1. *READ InitialList = (L), ListProp*
2. TRANSMIT AddListElement-Request,
   'Object Identifier' = *(L)*,
   'Property Identifier' = ListProp,
   'List of Elements' = (two or more elements to be added to the list with the second element having the wrong datatype)
3. *IF (Protocol_Revision is present and Protocol_Revision ≥ 7) THEN*
   RECEIVE AddListElement-Error,
   Error Class = PROPERTY,
   Error Code = INVALID_DATATYPE
   'First Failed Element' = 2
   ELSE
   RECEIVE AddListElement-Error,
   Error Class = SERVICES,
   Error Code = INVALID_PARAMETER_DATATYPE
   'First Failed Element' = 2
   | *(AddListElement-Error,*
   Error Class = PROPERTY,
   Error Code = INVALID_DATATYPE)
4. VERIFY (L), ListProp = InitialList

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

[Change **Clause 9.15.2.2**, p 247]

9.15.2.2 A RemoveListElement Failure Part Way Through a List

Purpose: To verify the ability of the IUT to respond to a RemoveListElement service request to remove multiple elements from a list where one of the elements cannot be removed. Upon failure, the RemoveListElement service should leave the list unchanged.

Test Steps:

~~1. TRANSMIT ReadProperty-Request,~~
~~'Object Identifier' = L,~~
~~'Property Identifier' = ListProp~~
~~2. RECEIVE ReadProperty-ACK,~~
~~'Object Identifier' = L,~~
~~'Property Identifier' = ListProp,~~
~~'Property Value' = (any valid value referred to as "InitialList" below)~~

1. *READ InitialList = (L), ListProp*
2. TRANSMIT RemoveListElement-Request,
   'Object Identifier' = *(*L*)*,
   'Property Identifier' = *(*ListProp*),*
   'List of Elements' = (one element from InitialList, followed by an element of the correct datatype that is not in InitialList, followed by one or more elements from InitialList)
3. *IF (Protocol_Revision is present and Protocol_Revision ≥ 7) THEN*
   RemoveListElement-Error,
   Error Class = *PROPERTY,*
   Error Code = *INVALID_DATA_TYPE*
   'First Failed Element' = *2*
   ELSE
   RECEIVE RemoveListElement-Error,
   Error Class = SERVICES | PROPERTY,
   Error Code = OTHER
   'First Failed Element' = 2
   | *(RemoveListElement-Error,*
   Error Class = *PROPERTY,*
   Error Code = *INVALID_DATA_TYPE*
   'First Failed Element' = *2)*
4. VERIFY (L), ListProp = InitialList

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-11 Relax ReadPropertyMultiple Error Test.

Rationale

Allow the IUT to return a ComplexACK with the error indicated instead of a BACnet-Error-PDU.

[Change **Clause 9.20.1.1**, p 262]

9.20.2.1 Reading a Single, Unsupported Property from a Single Object

Purpose: To verify the ability to correctly execute a ReadPropertyMultiple service request for which the 'List of Read Access Specifications' contains specifications for a single unsupported property.

…

Test Steps:

1. TRANSMIT ReadPropertyMultiple-Request,
   'Object Identifier' = Object1 | Object2,
   'Property Identifier' = (any property, P1, that is not supported in the selected object)
2. RECEIVE BACnet-Error-PDU,
   'Error Class' = PROPERTY,
   'Error Code' = UNKNOWN_PROPERTY
   | *(ReadPropertyMultiple-ACK,*
   'Object Identifier' = (the object identifier from step 1),
   'Property Identifier' = P1,
   'Error Class' = PROPERTY,
   'Error Code' = UNKNOWN_PROPERTY)

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-12 Add Test for Unicast Who-Is.

Rationale

Add a test that tests unicast Who-Is initiation.

[Add new **Clause 8.34.X1,** p. 178]

8.34.X1 Who-Is Request with no Device Instance Range

Purpose: To verify that the IUT can initiate unicast Who-Is service requests with no device instance range. If the IUT cannot be caused to issue a Who-Is request of this form, then this test shall be omitted.

Test Steps:

1. RECEIVE
   DESTINATION = TD,
   SOURCE = IUT,
   Who-Is-Request

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-13 Revise Unknown Network Layer Message Test.

Rationale

Change test 10.2.2.7.2 to match the result of IC-2004-12.

[Change **Clause 10.2.2.7.2**, p. 323]

10.2.2.7.2 Unknown Network Layer Message Type

Purpose: To verify that the IUT will reject a network layer message ~~with an unknown message type~~ *directed to the IUT that contains an unknown message type in the range of message types reserved for use by ASHRAE.*

Test Steps:

1. TRANSMIT PORT A,
   DESTINATION = IUT,
   SOURCE = ~~D1A~~ *TD*,
   Message Type = (any value ~~from X'0A' to X'7F~~ *in the range reserved for use by ASHRAE that is undefined in the protocol revision claimed by the device*)
2. RECEIVE PORT A,
   DESTINATION = ~~D1A~~ *TD*,
   SOURCE = IUT,
   Reject-Message-To-Network,
   Reject Reason = 3 (unknown network layer message type),
   DNET = ~~1~~ *any value*

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135.1-2009*i*-14 Add New Trend Log Tests.

Rationale

A collection of new Trend Log tests are added to expand upon the set of testable functionality and existing tests are changed to work on all log object types, where applicable (Trend Log, Trend Log Multiple, and Event Log).

[Add new **Clause 7.3.2.24.X1**, p. 105]

7.3.2.24.X1 Log-Status Test

Dependencies: ReadRange Service Execution Tests, 9.21; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clause: 12.25.14.

Purpose: To verify proper logging of log-disabled and buffer-purged events.

Test Concept: The buffer is cleared. Then the Enable property is changed and it is verified that the Record_Count property is changed and it is verified that the status entry is made correctly in the Log_Buffer. The Record_Count is also set to zero while the Enable property is FALSE and it is verified that the buffer-purged event is recorded into the Log_Buffer.

Test Configuration: The Trend Log, O1, is configured to acquire data by whatever means available. Configure the logging such that the entire test may be run without the trend buffer overflowing.

Test Steps:

1. WRITE Enable = FALSE
2. WRITE Record_Count = 0
3. VERIFY Record_Count = 1
4. TRANSMIT ReadRange
   - 'Object Identifier' = O1,
   - 'Property Identifier' = Log_Buffer,
   - 'Reference Index' = 1,
   - 'Count' = 1
5. RECEIVE ReadRange-Ack
   - 'Object Identifier' = O1,
   - 'Property Identifier' = Log_Buffer,
   - 'Result Flags' = (True, True, False),
   - 'Item Count' = 1
   - 'Item Data' = ((a buffer purged record))
6. WRITE Enable = TRUE
7. WRITE Enable = FALSE
8. TRANSMIT ReadRange
   - 'Object Identifier' = O1,
   - 'Property Identifier' = Log_Buffer,
   - 'Reference Index' = 1,
   - 'Count' = 2
9. RECEIVE ReadRangeAck
   - 'Object Identifier' = O1,
   - 'Property Identifier' = Log_Buffer,
   - 'Result Flags' = (True, False, False),
   - 'Item Count' = 2
   - 'Item Data' = ((a buffer purged record), (a log-enable record))
10. TRANSMIT ReadRange

© 2011 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc. (www.ashrae.org). For personal use only.
Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

‘Object Identifier’ = O1,
‘Property Identifier’ = Log_Buffer,
‘Reference Time’ = (2154-12-31, 23:59:59.99),
‘Count’ = -1

11. RECEIVE ReadRangeAck
‘Object Identifier’ = O1,
‘Property Identifier’ = Log_Buffer,
‘Result Flags’ = (False, True, False),
‘Item Count’ = 1
‘Item Data’ = ((a log-disable record))

[Add new **Clause 7.3.2.24.X2**, p. 105]

7.3.2.24.X2 Time_Change Test

Dependencies: ReadRange Service Execution Tests, 9.21; (TimeSynchronization Service Execution Tests, 9.30 or UTCTimeSynchronization Service Execution Tests, 9.31)

BACnet Reference Clause: 12.25.14.

Purpose: To verify proper logging of time-change events in the log buffer

Test Concept: Change the clock in the device and verify that a record is logged indicating the number of seconds that the clock changed by or indicating zero if unknown. This test shall be skipped if the device does not support the Local_Time property in the device object or there is no way to change the time in the device.

Test Configuration: The log is configured to acquire data by whatever means available. The Log_Buffer should be cleared such that the Record_Count is 0. Configure the logging such that the entire test may be run without the trend buffer overflowing.

Test Steps:

1. WRITE Enable = FALSE
2. WRITE Record_Count = 0
3. READ currentTime = (Device Object of device that contains the log object), Local_Time
4. WRITE Enable = TRUE
5. MAKE(the time change on the device by deltaTime where deltaTime >= 1 hour)
6. WRITE Enable = FALSE
7. READ N = Record_Count
8. REPEAT X = (N down through 1) DO {
TRANSMIT ReadRange
‘Object Identifier’ = O1,
‘Property Identifier’ = Log_Buffer,
‘Reference Index’ = X,
‘Count’ = 1
RECEIVE ReadRangeAck
‘Object Identifier’ = O1,
‘Property Identifier’ = Log_Buffer,
‘Result Flags’ = (False, True, False),
‘Item Count’ = 1,
‘Item Data’ = ((a record. If the record is a time-change record, save the timestamp into TS and the time-change value into TC))
}
9. CHECK (TC ~= deltaTime)
10. CHECK (TS ~= currentTime + deltaTime)