BACnet® TESTING LABORATORIES ADDENDA

Addendum e to BTL Test Package 15.1

Revision 1.0
Revised 4/21/2018

Approved by the BTL Working Group on July 13, 2018;
Approved by the BTL Working Group Voting Members on August 7, 2018;
Published on August 9, 2018

[This foreword and the "Overview" on the following pages are not part of this Test Package. They are merely informative and do not contain requirements necessary for conformance to the Test Package.]

FOREWORD

The purpose of this addendum is to present current changes being made to the BTL Test Package. These modifications are the result of change proposals made pursuant to the continuous maintenance procedures and of deliberations within the BTL-WG Committee. The changes are summarized below.

BTL-15.1e-1: Relax AcknowledgeAlarm Service Test, p. 2 [wID0986, BTLWG-70, BTL-CR-0388]
BTL-15.1e-2: Removes One-Hop tests from NAT BBMD Test Plan, p. 5 [BTLWG-91, BTL-CR-0399]
BTL-15.1e-3: Corrects Directed to a Remote Device Test, p. 6 [BTLWG-220, wID0980]
BTL-15.1e-4: Updates Test Conditionality for Alarm Acknowledgement tests, p. 7 [BTLWG-243, BTL-CR-0409]
BTL-15.1e-5: Modifies Event_Detection_Enable to match specification, p. 8 [BTLWG-320, BTL-CR-0417]
BTL-15.1e-6: Relax Date-Pattern Properties Test, p. 9 [BTL6WG-321, BTL-CR-0418]
BTL-15.1e-7: Modify Active_COV_Subscriptions Subscribe COV Test, p. 10 [BTL6WG-366, wID0964]
BTL-15.1e-8: Correct the TimeSynchronization and UTC TimeSynchronization Tests, p. 13 [BTL6WG-367, wID1044]
BTL-15.1e-9: Clarify when the ReadRange test on unknown property can be skipped, p. 16 [BTL6WG-368, wID1059]
BTL-15.1e-10: Add Test to Ignore Confirmed Broadcasts, p. 17 [BTL6WG-369, wID1115]

In the following document, language to be added to existing clauses within the BTL Test Package 15.1 is indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are proposed to be added, plain type is used throughout

In addition, changes to BTL Specified Tests also contain a yellow highlight to indicate the changes made by this addendum.

When this addendum is applied, all highlighting will be removed. Change markings on tests will remain to indicate the difference between the new test and an existing 135.1 test. If a test being modified has never existed in 135.1, the applied result should not contain any change markings. When this is the case, square brackets will be used to describe the changes required for this test.

Each addendum can stand independently unless specifically noted via dependency within the addendum. If multiple addenda change the same test or section, each future released addendum that changes the same test or section will note in square brackets whether or not those changes are reflected.

BTL-15.1e-1: Relax AcknowledgeAlarm Service Test

Overview:
Clarification Request BTL-CR-0388 agrees the AcknowledgeAlarm test is too strict and should be modified.

Changes:
The change here allows the 'To State' to be any off-normal state.

[In BTL Specified Tests, add the following test.]

8.1 AcknowledgeAlarm Service Initiation Tests

Dependencies: None.

BACnet Reference Clause: 13.5.

Purpose: To verify that the IUT is capable of acknowledging alarms and events that are reported to the IUT via the ConfirmedEventNotification and UnconfirmedEventNotfication services.

Configuration: For this test, the tester shall choose 1 object, O1, in the TD, which is configured to send event notifications to the IUT. The tester places O1 into an alarm state such that the transition requires an acknowledgment.

Test Steps:

1. TRANSMIT ConfirmedEventNotification-Request | UnconfirmedEventNotification-Request,
   'Subscriber Process Identifier' = (a value acceptable to the IUT configured in the Notification Class object for the IUT),
   'Initiating Device Identifier' = TD,
   'Event Object Identifier' = O1,
   'Time Stamp' = (any valid value, T1),
   'Notification Class' = (the value configured in O1),
   'Priority' = (any value selected by the TD),
   'Event Type' = (any value selected by the TD),
   'Notify Type' = ALARM | EVENT,
   'AckRequired' = TRUE,
   'From State' = (any valid value),
   'To State' = (any valid value, S1),
   'Event Values' = (any event values appropriate to the event type)
2. IF (the ConfirmedEventNotification choice was selected) THEN
   RECEIVE BACnet-SimpleACK-PDU
3. MAKE (the IUT acknowledge O1)
4. RECEIVE AcknowledgeAlarm-Request,
   'Acknowledging Process Identifier' = (any process identifier),
   'Event Object Identifier' = O1,
   'Event State Acknowledged' = S1, *or OFFNORMAL if S1 is an off-normal state*
   'Time Stamp' = T1,
   'Acknowledgement Source' = (any valid value),
   'Time of Acknowledgement' = (any valid value)
5. TRANSMIT BACnet-SimpleACK-PDU

[In BTL Test Plan, change the reference to BTL for the indicated test]

5.4Alarm and Event - ACK - A

5.4.1 Base Requirements

Base requirements must be met by any IUT claiming conformance to this BIBB.

***BTL ~~135.1-2013~~ -* 8.1 - AcknowledgeAlarm Service Initiation Tests**	
Test Method	Manual
Configuration	As per ***BTL Specified Tests***. ***~~ASHRAE 135.1-2013.~~***
Test Conditionality	Must be executed once to acknowledge a ConfirmedEventNotification, and again to acknowledge an UnconfirmedEventNotification.
Test Directives	
Testing Hints	
Notes & Results	

5.4.2 Generates AcknowledgeAlarm Requests with Acknowledge Timestamps of the BACnetDateTime Form

The IUT can generate AcknowledgeAlarm requests with a Time of Acknowledgement of the BACnetDateTime form.

BTL ~~135.1-2013~~ - 8.1 - AcknowledgeAlarm Service Initiation Tests	
Test Method	Manual
Configuration	As per ***BTL Specified Tests***. ***~~ASHRAE 135.1-2013.~~***
Test Conditionality	Must be executed. Verify that the Time of Acknowledgement parameter is of the BACnetDateTime form.
Test Directives	
Testing Hints	
Notes & Results	

5.4.3 Generates AcknowledgeAlarm Requests with Acknowledge Timestamps of the Time Form

The IUT can generate AcknowledgeAlarm requests with a Time of Acknowledgement of the Time form.

BTL ~~135.1-2013~~ - 8.1 - AcknowledgeAlarm Service Initiation Tests	
Test Method	Manual
Configuration	As per ***BTL Specified Tests***. ***~~ASHRAE 135.1-2013.~~***
Test Conditionality	Must be executed. Verify that the Time of Acknowledgement parameter is of the Time form.
Test Directives	
Testing Hints	
Notes & Results	

5.4.4 Generates AcknowledgeAlarm Requests with Acknowledge Timestamps of the Sequence Number Form

The IUT can generate AcknowledgeAlarm requests with a Time of Acknowledgement of the Sequence Number form.

BTL ~~135.1-2013~~ - 8.1 - AcknowledgeAlarm Service Initiation Tests	
Test Method	Manual
Configuration	As per ***BTL Specified Tests***. ***~~ASHRAE 135.1-2013.~~***
Test Conditionality	Must be executed. Verify that the Time of Acknowledgement parameter is of the Sequence Number form.
Test Directives	

Testing Hints	
Notes & Results	

BTL-15.1e-2: Removes One-Hop tests from NAT BBMD Test Plan

Overview:
BTL-CR-0399 clarified that the definition for NAT BBMD in 135-2012j does not allow One-hop Distribution.

Changes:
The Test Plan section for BBMD Supports Network Address Translation is modified to remove the One-hop Distribution tests.

[In BTL Test Plan, remove sections highlighted below.]

9.4.6 BBMD Supports Network Address Translation

The IUT is capable of operating behind a router providing Network Address Translation as described in addendum 135-2008*o*-1.

~~**BTL - 14.2.1.1 - Execute Forwarded-NPDU (One-hop Distribution)**~~		
	~~**Test Method**~~	~~Manual~~
	~~**Configuration**~~	~~As per ***BTL Specified Tests***.~~
	~~**Test Conditionality**~~	~~Must be executed.~~
	~~**Test Directives**~~	~~Internet Routers and the IUT shall be configured for NAT.~~
	~~**Testing Hints**~~	
	~~**Notes & Results**~~	
~~**BTL - 14.2.2.1 - Execute Original-Broadcast-NPDU (One-hop Distribution)**~~		
	~~**Test Method**~~	~~Manual~~
	~~**Configuration**~~	~~As per ***BTL Specified Tests***.~~
	~~**Test Conditionality**~~	~~Must be executed.~~
	~~**Test Directives**~~	~~Internet Routers and the IUT shall be configured for NAT.~~
	~~**Testing Hints**~~	
	~~**Notes & Results**~~	
BTL - 14.7.1.2 - Broadcast Message from Directly Connected IP Subnet (Two-hop Distribution)		
	Test Method	Manual
	Configuration	As per ***BTL Specified Tests***.
	Test Conditionality	Must be executed.
	Test Directives	Internet Routers and the IUT shall be configured for NAT.
	Testing Hints	
	Notes & Results	
BTL - 14.7.2.2 - Broadcast Message Forwarded by a Peer BBMD (Two-hop Distribution)		
	Test Method	Manual
	Configuration	As per ***BTL Specified Tests***.
	Test Conditionality	Must be executed.
	Test Directives	Internet Routers and the IUT shall be configured for NAT.
	Testing Hints	
	Notes & Results	
BTL - 14.7.3.2 - Broadcast Message From a Foreign Device (Two-hop Distribution)		
	Test Method	Manual
	Configuration	As per ***BTL Specified Tests***.
	Test Conditionality	Must be executed.
	Test Directives	Internet Routers and the IUT shall be configured for NAT.
	Testing Hints	
	Notes & Results	

BTL-15.1e-3: Corrects Directed to a Remote Device Test

Overview:
Test 135.1-2013 9.33.2.3 incorrectly allows for a Local Broadcast response instead of a Remote Broadcast. This correction was made in BTL Test plan 5.0 but was not carried over to the 135.1 document. In addition, the BACnet standard (per addendum 2008q) now allows the IUT to send a unicast response.

Changes:
The purpose statement has been modified to match the wording used in 9.33.2.1 (Global broadcast) and 9.33.2.2 (Remote broadcast).

[In BTL Specified Tests, derive a changed test from 135.1-2013.]

9.33.2.3 General Inquiry, Directed to a Remote Device

Purpose: To verify ~~that the IUT responds with an I-Am service that is of the form global broadcast, remote broadcast or unicast~~ *the ability of the IUT to recognize the origin of a Who-Is service request, directed to the IUT, and respond such that the device originating the request receives the response.*

Test Steps:

1. TRANSMIT
   DESTINATION = IUT,
   SNET = (any remote network number),
   SADR = (any MAC address valid for the specified network),
   Who-Is-Request
2. WAIT ~~**Internal Processing**~~ ***Unconfirmed Response*** **Fail Time**
3. RECEIVE
   DESTINATION = GLOBAL BROADCAST | ~~LOCAL BROADCAST~~ *REMOTE BROADCAST (to the network specified by SNET in step 1)* | *TD*
   I-Am-Request,
   'I Am Device Identifier' = (the IUT's Device object),
   'Max APDU Length Accepted' = (the value specified in the EPICS),
   'Segmentation Supported' = (the value specified in the EPICS),
   'Vendor Identifier' = (the identifier registered for this vendor)

BTL-15.1e-4: Updates Test Conditionality for Alarm Acknowledgement tests

Overview:
As CR-0409 made clear the test can only be applied when the IUT supports High-Limit or Low-Limit transitions.

Changes:
The Test Conditionality is changed to make it clearer, and removed the wording about Revision 5 as the current Test Plan only applies to revision 12 and higher.

[In BTL Test Plan, modify Test Conditionality of section 5.5.1 for test 9.1.1.X3]

BTL - 9.1.1.X3 - Successful Alarm Acknowledgment of Confirmed Event Notifications when 'To State' is either High-Limit or Low-Limit, Revision 5 and higher only		
	Test Method	Manual
	Configuration	As per ***BTL Specified Tests***
	Test Conditionality	If the IUT supports LOW_LIMIT or HIGH_LIMIT transitions ~~revision 5 or higher~~ this test must be executed.
	Test Directives	
	Testing Hints	
	Notes & Results	

BTL-15.1e-5: Modifies Event_Detection_Enable to match specification

Overview:
During discussion of CR-417 it was noted that 7.3.1.X9.1 as written goes against the standard's position that Event_Detection_Enable is not expected to be modified during normal operation. Because of this position, there is no mandate on the operation of the device when this property is changed from TRUE to FALSE.

Changes:
This proposal changes the test to only verify the conditions that are expected when Event_Detection_Enable is FALSE and does not perform testing related to actively changing the property.

[In BTL Specified Tests, replace existing 7.3.1.X9.1 with the new version below.]

7.3.1.X9.1 Event_Detection_Enable Inhibits Event Generation

Reason for Change: New functionality added with Addendum 135-2010af. This test does not exist in 135.1-2013.

Purpose: To verify that Event_Detection_Enable disables event detection.

Test Concept: Select an event generating object, O1, that supports event reporting. Verify that Event_State is NORMAL and Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts are equal to their respective initial conditions, as mandated in the standard. If possible, make a condition exist that would cause a transition if event reporting were enabled and observe that no notification messages are transmitted.

Configuration Requirements: O1 is configured with Event_Detection_Enable set to FALSE. DELAY shall represent the time delay appropriate to the transition being tested (i.e. Time_Delay for TO_OFFNORMAL, 0 for TO_FAULT and FAULT to NORMAL transitions, and either Time_Delay or Time_Delay_Normal for TO_NORMAL). For this test, NO_TS equals a BACnetDateTime with all unspecified values, a BACnet Time with all unspecified values, or a sequence number of 0.

Test Steps:
1. VERIFY Event_Detection_Enable = FALSE
2. VERIFY Event_State = NORMAL
3. VERIFY Acked_Transitions = (T,T,T)
4. VERIFY Event_Time_Stamps = [NO_TS , NO_TS , NO_TS]
5. IF the Event_Message_Texts property exists THEN
   VERIFY Event_Message_Texts = ['', '', '']
6. MAKE (a condition exist which would cause O1 to transition, if Event_Detection_Enable were TRUE)
7. WAIT DELAY + Notification Fail Time
8. CHECK (that the IUT did not send any event notifications for O1)
9. VERIFY Event_State = NORMAL
10. VERIFY Acked_Transitions = (T,T,T)
11. VERIFY Event_Time_Stamps = [NO_TS, NO_TS, NO_TS]
12. IF the Event_Message_Texts property exists THEN
    VERIFY Event_Message_Texts = ['', '', '']

BTL-15.1e-6: Relax Date-Pattern Properties Test

Overview:
In CR-418, it was noted that a device arrived in the lab which updates the day-of-week to be consistent with the other fields in a date value when it is the only unspecified field in the date value. The BTL-WG decided that this was not problematic behavior.

Changes:
The test has been changed to allow it and still test support for unspecified day-of-week.

[In BTL Specified Tests, replace existing Date Pattern Test]

7.2.X1 Date Pattern Properties Test

Reason for Change: Addendum 135-2001a-1 adds odd and even month support, and last-day-of-the-month special value. Addendum 135-2008h.8h-8 adds odd and even day support. Addendum 135-2008acac-1 clarifies when wildcards are allowed in dates and times. Test does not exist in 135.1-2013.

Purpose: To verify that the property being tested accepts special date field values.

Test Concept: The property being tested, P1, is written with each of the special date field values to ensure that the property accepts them. A date, D1, is selected which is within the date range that the IUT will accept for the property. The value, written to the property is the date D1 with one of its fields replaced with one of the date special values. If the property is a complex datatype, the other fields in the value shall be set within the range accepted by the IUT. The list of Specials comes from the Chapter 21 Application Types section on Date. *The day-of-week field is redundant information and can be regenerated from the other fields. In order to not fail devices which always ensure this field is consistent with the other fields in the date value, the use of an unspecified day of week is always tested in conjunction with another pattern value.*

Test Steps:

1. IF (Protocol_Revision is not present or Protocol_Revision < 4)
   Specials = (year unspecified, month unspecified, day of month unspecified~~, day of week unspecified~~)
   ELSE IF (Protocol_Revision >= 4 and Protocol_Revision < 10)
   Specials = (year unspecified, month unspecified, day of month unspecified, ~~day of week unspecified,~~ odd months, even months, last day of month)
   ELSE
   Specials = (year unspecified, month unspecified, day of month unspecified, ~~day of week unspecified,~~ odd months, even months, last day of month, even days, odd days)
2. REPEAT SV = (each value in Specials) DO {
   IF SV <> day of week unspecified THEN
   V1 = D1 updated with the value SV
   ELSE
   V1 = D1 updated with the value SV and any other value from Specials
   WRITE P1 = (~~D1 updated with the value SV~~*V1*)
   VERIFY P1 = (~~D1 updated with the value SV~~*V1*)
   }

Notes to Tester: if P1 is an array, then an array index shall be provided in the WRITE and VERIFY operations.

BTL-15.1e-7: Modify Active_COV_Subscriptions Subscribe COV Test

Overview:
SSPC has clarified via official **Interpretation** that the COV_Increment shall be included in an Active_COV_Subscriptions entry if the subscribed to property is numeric and a COV increment was included in the subscription request. The COV_Increment shall not be included in an Active_COV_Subscriptions entry if the subscribed to property is non-numeric.

It is a local matter whether or not the COV_Increment is included in an Active_COV_Subscriptions entry if the subscribed to property is numeric and no COV increment was included in the subscription request.

Changes:
Also the test needs to restore Steps 15 through 21. The existing test 135.1-2013 - 7.3.2.10.1 was accidentally edited into 135.1-2013 from 135.1-2009d-2, without including steps 15 through 21.

[In BTL Specified Tests, deriving from existing test 135.1-2013 - 7.3.2.10.1, correct by giving the representation of the full test. This goes in an errata, not into an addenda.]

7.3.2.10.1 Active_COV_Subscriptions SubscribeCOV Test

Reason for Change: IC135-2012-18 ruled that the increment shall be in the Active_COV_Subscriptions property value if the property is not numeric; present if a valid Increment was provided in the subscription; and optionally present otherwise

Purpose: This test case verifies that the IUT correctly updates the Active_COV_Subscriptions property when COV subscriptions are created, cancelled and timed-out using SubscribeCOV.

Test Concept: *INC_1, INC_2, and INC_3 are each not present if the property is not numeric; present if a valid Increment was provided in the subscription; and optionally present otherwise.*

Configuration Requirements: In this test, the tester shall choose three standard objects, O_1, O_2, and O_3, for which the device supports SubscribeCOV. O_1, O_2, and O_3 are not required to refer to different objects. The tester shall also choose three non-zero unique process identifiers, P_1, P_2, and P_3, and three non-zero lifetimes L_1, L_2 and L_3. Lifetime L_1 shall be long enough to allow the initial part of the test to run through to step 14. Lifetimes L_2 and L_3 shall be long enough for the whole test to be completed without expiring.

The IUT shall start the test with no entries in its Active_COV_Subscriptions property.

Test Steps:

1. TRANSMIT SubscribeCOV-Request,
   'Subscriber Process Identifier' = P_1,
   'Monitored Object Identifier' = O_1,
   'Issue Confirmed Notifications' = TRUE,
   'Lifetime' = L_1
2. RECEIVE BACnet-SimpleACK-PDU
3. BEFORE **Notification Fail Time**
   RECEIVE ConfirmedCOVNotification-Request,
   'Subscriber Process Identifier' = P_1,
   'Initiating Device Identifier' = IUT,
   'Monitored Object Identifier' = O_1,
   'Time Remaining' = (a value approximately equal to L_1),
   'List of Values' = (values appropriate to the object type of the monitored object)
4. TRANSMIT BACnet-SimpleACK-PDU
5. *IF P1 is numeric*
   VERIFY Active_COV_Subscriptions = {{ {TD, P_1}, { O_1, Present_Value }, TRUE, (a value less than L_1),(*INC_1 : not present or* a valid Increment ~~if the property is REAL~~)
}}
   Else
   VERIFY Active_COV_Subscriptions = {{ {TD, P_1}, { O_1, Present_Value }, TRUE, (a value less than L_1), (INC_1: not present)}}
6. TRANSMIT SubscribeCOV-Request,
   'Subscriber Process Identifier' = P_2,

'Monitored Object Identifier' = O_2,
'Issue Confirmed Notifications' = FALSE,
'Lifetime' = L_2

7. RECEIVE BACnet-SimpleACK-PDU
8. BEFORE **Notification Fail Time**
   RECEIVE UnconfirmedCOVNotification-Request,
   'Subscriber Process Identifier' = P_2,
   'Initiating Device Identifier' = IUT,
   'Monitored Object Identifier' = O_2,
   'Time Remaining' = (a value approximately equal to L_2),
   'List of Values' = (values appropriate to the object type of the monitored object)
9. VERIFY Active_COV_Subscriptions = {{{TD, P_1}, { O_1, Present_Value }, TRUE, (a value less than L_1), *INC_1* ~~(a valid Increment if the property is REAL)~~ },
   {{TD, P_2,}, {O_2, Present_Value}, FALSE, (a value less than L_2), (*INC_2 : not present if the property is not numeric; present if* a valid Increment *was provided in the subscription; optionally present otherwise*~~if the property is REAL~~) }}
10. TRANSMIT SubscribeCOV-Request,
    'Subscriber Process Identifier' = P_3,
    'Monitored Object Identifier' = O_3,
    'Issue Confirmed Notifications' = FALSE,
    'Lifetime' = L_3
11. RECEIVE BACnet-SimpleACK-PDU
12. BEFORE **Notification Fail Time**
    RECEIVE UnconfirmedCOVNotification-Request,
    'Subscriber Process Identifier' = P_3,
    'Initiating Device Identifier' = IUT,
    'Monitored Object Identifier' = O_3,
    'Time Remaining' = (a value approximately equal to L_3),
    'List of Values' = (values appropriate to the object type of the monitored object)
13. *IF P3 is numeric*
    VERIFY Active_COV_Subscriptions = {{{TD, P_1}, {O_1, Present_Value}, TRUE, (a value less than L_1), *INC_1* ~~(a valid Increment if the property is REAL)~~},
    {{TD, P_2,}, {O_2, Present_Value}, FALSE, (a value less than L_2), *INC_2* ~~(a valid Increment if the property is REAL)~~},
    {{TD, P_3}, {O_3, Present_Value}, FALSE, (a value less than L_3), *INC_3: not present or* (a valid Increment ~~if the property is REAL)~~}}

    Else
    VERIFY Active_COV_Subscriptions = {{{TD, P_1}, {O_1, Present_Value}, TRUE, (a value less than L_1), INC_1 },
    {{TD, P_2,}, {O_2, Present_Value}, FALSE, (a value less than L_2), INC_2 },
    {{TD, P_3}, {O_3, Present_Value}, FALSE, (a value less Than L_3), (INC_3: not present)}}

14. WAIT L_1 + the IUT's timer granularity
15. *VERIFY Active_COV_Subscriptions = {{{TD, P_2 }, { O_2 , Present_Value}, FALSE, (a value less than L_2),INC_2 (a valid Increment if the property is REAL)},*
    {{TD, P_3 }, {O_3 , Present_Value}, FALSE, (a value less than L_3),INC_3 (a valid Increment if the property is REAL)}}
16. *TRANSMIT SubscribeCOV-Request,*
    'Subscriber Process Identifier' = P_3 ,
    'Monitored Object Identifier' = O_3
17. *RECEIVE BACnet-SimpleACK-PDU*
18. *VERIFY Active_COV_Subscriptions = {{{TD, P_2}, { O_2, Present_Value}, FALSE, (a value less than L_2), INC_2 (a valid Increment if the property is REAL) }}*
19. *TRANSMIT SubscribeCOV-Request,*
    'Subscriber Process Identifier' = P_2 ,
    'Monitored Object Identifier' = O_2

20. RECEIVE BACnet-SimpleACK-PDU
21. VERIFY Active_COV_Subscriptions = { }

[In BTL Test Plan, change the reference to BTL for the indicated test]

4.10 Data Sharing - COV - B

4.10.1 Base Requirements

Base requirements must be met by any IUT claiming conformance to this BIBB.

BTL~~135.1-2013~~ - **7.3.2.10.1 Active_COV_Subscriptions SubscribeCOV Test**		
	Test Method	Manual
	Configuration	As per ***BTL Specified Tests~~ASHRAE 135.1-2013~~***.
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
	Notes & Results	

BTL-15.1e-8: Correct the TimeSynchronization and UTCTimeSynchronization Tests

Overview:
The TimeSynchronization includes properties that are not required. Upon review of the related tests, they were changed to simplify the testing.

Changes:
The TimeSynchronization and UTCTimeSynchronization tests were modified to correct and simply them.

[In **BTL Specified Tests, modify the existing test.-9.30.1.1**]

9.30.1.1 *TimeSynchronization* Local Broadcast

Reason for change: UTC_Offset and Daylight_Savings_Status are optional properties that are only required for the UTCTimeSynchronization service.
In some test steps, TRANSMIT was misused instead of READ and VERIFY.

Purpose: To verify that the IUT resets its local time and date in response to a local broadcast TimeSynchronization service request.

Test Steps:

~~1. TRANSMIT ReadProperty-Request,~~
~~'Object Identifier' = (the IUT's Device object),~~
~~'Property Identifier' = Local_Date~~
~~2. RECEIVE ReadProperty-ACK,~~
~~'Object Identifier' = (the IUT's Device object),~~
~~'Property Identifier' = Local_Date,~~
~~'Property Value' = (any valid date referred to as "InitialDate" below)~~
~~3. TRANSMIT ReadProperty-Request,~~
~~'Object Identifier' = (the IUT's Device object),~~
~~'Property Identifier' = Local_Time~~
~~4. RECEIVE ReadProperty-ACK,~~
~~'Object Identifier' = (the IUT's Device object),~~
~~'Property Identifier' = Local_Time,~~
~~'Property Value' = (any valid time referred to as "InitialTime" below)~~
~~5. TRANSMIT ReadProperty-Request,~~
~~'Object Identifier' = (the IUT's Device object),~~
~~'Property Identifier' = UTC_Offset~~
~~6. RECEIVE ReadProperty-ACK,~~
~~'Object Identifier' = (the IUT's Device object),~~
~~'Property Identifier' = UTC_Offset,~~
~~'Property Value' = (any valid offset referred to as "InitialUTC_Offset" below)~~
~~7. TRANSMIT ReadProperty-Request,~~
~~'Object Identifier' = (the IUT's Device object),~~
~~'Property Identifier' = Daylight_Savings_Status~~
~~8. RECEIVE ReadProperty-ACK,~~
~~'Object Identifier' = (the IUT's Device object),~~
~~'Property Identifier' = Daylight_Savings_Status,~~
~~'Property Value' = (any valid status referred to as "InitialDaylight_Savings_Status" below)~~
~~5. TRANSMIT~~
~~DA = LOCAL BROADCAST,~~
~~SA = TD,~~
~~BACnet-Unconfirmed-Request-PDU,~~
~~'Service Choice' = TimeSynchronization-Request,~~
~~date = (any date other than InitialDate),~~
~~time = (any time that does not correspond to InitialTime)~~
~~6. TRANSMIT ReadProperty-Request,~~
~~'Object Identifier' = (the IUT's Device object),~~
~~'Property Identifier' = Local_Date~~

~~7. RECEIVE ReadProperty-ACK,~~
~~'Object Identifier' = (the IUT's Device object),~~
~~'Property Identifier' = Local_Date,~~
~~'Property Value' = (the date specified in step 5)~~
~~8. TRANSMIT ReadProperty-Request,~~
~~'Object Identifier' = (the IUT's Device object),~~
~~'Property Identifier' = Local_Time~~
~~9. RECEIVE ReadProperty-ACK,~~
~~'Object Identifier' = (the IUT's Device object),~~
~~'Property Identifier' = Local_Time,~~
~~'Property Value' = (the time specified in step 5)~~

1. READ InitialDate = Local_Date
2. READ InitialTime = Local_Time
3. TRANSMIT
DA = LOCAL BROADCAST,
SA = TD,
BACnet-Unconfirmed-Request-PDU,
'Service Choice' = TimeSynchronization-Request,
date = NewDate: combined with NewTime is different than the InitialDate/InitialTime pair
time = NewTime; combined with NewDate is different than the InitialDate/InitialTime pair
4. VERIFY Local_Date = NewDate
5. VERIFY Local_Time ~= NewTime

Notes to Tester: Select date and time such that either one or both of them is different from initial date and time.

[In **BTL Specified Tests, modify the existing test-9.31.1.1**]

9.31.1.1 *UTCTimeSynchronization* Local Broadcast

Reason for change: UTC_Offset and Daylight_Savings_Status are needed here, as these optional properties are required for the UTCTimeSynchronization service. In some test steps, TRANSMIT was misused instead of READ and VERIFY.

Purpose: To verify that the IUT resets its local time and date in response to a local broadcast UTCTimeSynchronization service request.

Test Steps:

~~Test Steps: The test steps are identical to the steps in 9.30.1.1 except that in step 9 the UTCTimeSynchronization request is used and the date and time conveyed represent UTC.~~

~~Passing Results: The passing results are identical to 9.30.1.1 except that the date in step 9 shall be corrected for InitialUTC_Offset, and the time in step 13 shall be corrected for both Initial_UTC_Offset and Daylight_Savings_Status (as defined in BACnet 16.7.2).~~

1. READ InitialDate = Local_Date
2. READ InitialTime = Local_Time
3. TRANSMIT
DA = LOCAL BROADCAST,
SA = TD,
BACnet-Unconfirmed-Request-PDU,
'Service Choice' = UTCTimeSynchronization-Request,
date = NewUtcDate: combined with NewUtcTime and converted to local time is different than the InitialDate/InitialTime pair
time = NewUtcTime: combined with NewUtcDate and converted to local time is different than the InitialDate/InitialTime pair

4. *VERIFY Local_Date =* *(NewUtcDate converted to local date/time using UTC_Offset and Daylight_Saving_Status)*
5. *VERIFY Local_Time ~=* *(NewUtcTime converted to local date/time using UTC_Offset and Daylight_Saving_Status)*

Notes to Tester: Select date and time such that either one or both of them is different from initial date and time. The IUT may update the Daylight_Savings_Status during the execution of the UTCTimeSynchronization request.

[In **BTL Test Plan, add test directives to test key dates as par the changes made by the recent wID1028**]

8.8 Device Management - Time Synchronization - B

8.8.1 Base Requirements

Base requirements must be met by any IUT claiming conformance to this BIBB.

BTL - 9.30.1.1 - TimeSynchronization Local Broadcast		
	Test Method	Manual
	Configuration	As per ***BTL Specified Tests***.
	Test Conditionality	Must be executed.
	Test Directives	Ensure to test with the following date values: Feb 28, Feb 29, Mar 1 Dec 31, Jan 1 of the following year 28-Feb-2100 if IUT supports Dates in the year 2100
	Testing Hints	
	Notes & Results	

8.10 Device Management - Time Synchronization - B

8.10.1 Base Requirements

Base requirements must be met by any IUT claiming conformance to this BIBB.

BTL - 9.31.1.1 - UTCTimeSynchronization Local Broadcast		
	Test Method	Manual
	Configuration	As per ***BTL Specified Tests***.
	Test Conditionality	Must be executed.
	Test Directives	Values tested in the UTCTimeSynchronization-Request should include at least one which, in combination with the UTC_Offset, shall cause the result to cross midnight so that the resulting date is different from the date in the UTCTimeSynchronization-Request. Ensure to test with the following date values: Feb 28, Feb 29, Mar 1 Dec 31, Jan 1 of the following year 28-Feb-2100 if IUT supports Dates in the year 2100
	Testing Hints	
	Notes & Results	

BTL-15.1e-9: Clarify when the ReadRange test on unknown property can be skipped

Overview:

The existing test case states that, 'any list property not supported by the IUT', which does not specify if list property should be applicable for the object or it can be any list property e.g. (Analog Input----Alarm_Values)

Changes:
Add configuration requirements with text to explain this test can be skipped if all list properties of an object are supported.

[In BTL Specified Tests, modify test according to highlights below.]

9.21.2.1 Attempting to Read a Property That Does not Exist

Reason For Change: 135-2008u-3. Functionality for this test does not exist in 135.1.

Purpose: To verify the correct execution of the ReadRange service request when the requested property does not exist. This test is only applied to devices with a Protocol_Revision of 10 or higher.

Configuration Requirements: If all the list properties applicable for the object under testing are supported, then this test shall be skipped.

Test Steps:

1. TRANSMIT ReadRange-Request,
   'Object Identifier' = (any object that exists in the IUT),
   'Property Identifier' = (any list property *applicable for that object but* not supported by the IUT),
2. RECEIVE BACnet-Error-PDU,
   'Error Class' = PROPERTY,
   'Error Code' = UNKNOWN_PROPERTY

BTL-15.1e-10: Add Test to Ignore Confirmed Broadcasts

Overview:

The current test plan “BTL Test Plan-14.0.final.doc” does not include any test case related to Standard 135, clause 5.4.5.1, "ConfirmedBroadcastReceived" scenario.

Changes:
Add new test to cover the missing test case noted above.

[Add new test to BTL Specified Tests, as shown]

13.1.X3 Ignore Confirmed Broadcast Requests

Reason for Change: No existing test.

Purpose: This test case verifies that the IUT will quietly discard any Confirmed-Request-PDU, whose destination address is a multicast or broadcast address, received from the network layer.

Test Concept: The TD transmits the Confirmed-Request-PDU services whose destination address is a multicast or broadcast address. The IUT is required to silently drop the requests because it should only respond to unicast confirmed requests.

Test Steps:

1. TRANSMIT Any BACnet-Confirmed-Request-PDU,
   DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST,

2. CHECK (that the IUT does not send any packets in response to above Confirmed-Request-PDU)

[Add new test reference at end of existing section of BTL Test Plan, as shown]

2.1 Basic Functionality (applies to all BACnet Devices)

2.1.1 Base Requirements

BTL - 13.1.X13 - Ignore Confirmed Broadcast Requests		
	Test Method	Manual
	Configuration	As per ***BTL Specified Tests***.
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
	Notes & Results	