BACnet® TESTING LABORATORIES ADDENDA

Addendum Errata to BTL Test Package 20.0.1

Revision v5
Revised 11/11/2022

Published on December 22, 2023.

BACnet is a registered trademark of ASHRAE. ASHRAE does not endorse, approve or test products for compliance with ASHRAE standards. Compliance of listed products to the requirements of ASHRAE Standard 135 is the responsibility of BACnet International. BTL is a registered trademark of BACnet International.

[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-20.0.1 Errata-1: Correct typo in BTL-9.23.2.X9[BTLWG-1298]....2
BTL-20.0.1 Errata-2: Correct Tests 7.3.1.X41.Y2 [BTLWG-1300]....3
BTL-20.0.1 Errata-3: Correct Tests 7.3.2.X56.7 [BTLWG-1301]....4
BTL-20.0.1 Errata-4: Correct Test Reference in Test Plan Section 3.17.4 [BTLWG-1302]....5
BTL-20.0.1 Errata-5: Correct Test 8.5.17.X1.2 [BTLWG-1318]....6
BTL-20.0.1 Errata-6: Correct Tests 7.3.1.22.1 and 7.3.1.22.2 [BTLWG-1219]....7
BTL-20.0.1 Errata-7: Fix Typos in Test 9.1.2.1 and 9.1.2.5 [BTLWG-1327, BTLWG-1346, BTLWG-1349]....9
BTL-20.0.1 Errata-8: Fix Test 9.9.1.2 [BTLWG-1345, BTLWG-1347]....16
BTL-20.0.1 Errata-9: Advanced Modify BIBBs Needed Directives Update for Test 8.22.5 [BTLWG-1351]....18
BTL-20.0.1 Errata-10: Modify Test Plan Standard Reference [BTLWG-1359]....19
BTL-20.0.1 Errata-11: Correct Test Reference in Elevator Bibb Section [BTLWG-1358]....20
BTL-20.0.1 Errata-12: Correct Typos in Elevator Bibb Section [BTLWG-1364]....21
BTL-20.0.1 Errata-13: Fix Test 8.4.17.X9.15 [BTLWG-1362, BTLWG-1368]....22
BTL-20.0.1 Errata-14: Fix Test 8.4.X11 Event Values Reference [BTLWG-1361, BTLWG-1367]....23
BTL-20.0.1 Errata-13: Fix Reference in Network Forwarder Test [BTLWG-1363]....24

In the following document, language to be added to existing clauses within the BTL Test Package 20.0.1 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 contrast, 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-20.0.1 Errata-1: Correct typo in BTL-9.23.2.X9[BTLWG-1298]

Overview:
9.23.2.X9 Date Non-Pattern Properties Test using WritePropertyMultiple Service

Step 1:
'Object Identifier' = O1,

Step 2:
'Object Identifier' = Object1,

Changes:

Checklist Changes

None

Test Plan Changes

None

Specified Test Changes

[Modify Test 9.23.2.X9]

In step 3, change 'Object Identifier' = Object1, to 'Object Identifier = O1,

BTL-20.0.1 Errata-2: Correct Tests 7.3.1.X41.Y2 [BTLWG-1300]

Overview:
Tests:
7.3.1.X41.Y2 Blink-Warn WARN_OFF Command Test (BTL Specified Tests 20.0)
7.3.1.X41.Y3 Blink-Warn WARN_RELINQUISH Command Test (BTL Specified Tests 20.0)

First T1 = current local time then T2 = current local time, as a rusult T2 is bigger than T1.

In Step 10:
10. VERIFY Egress_Time ~= (T1 – T2) +/- Internal Processing Fail Time
…

Subtraction T1-T2 results in negative value because T1 < T2.

Should be
10. VERIFY Egress_Time ~= (T2 – T1) +/- Internal Processing Fail Time

Changes:

Checklist Changes

None

Test Plan Changes

Specified Test Changes

[Modify Test 7.3.1.X41.Y2]
Change Step 10 from:
10. VERIFY Egress_Time ~= (T1 – T2) +/- **Internal Processing Fail Time**

to
10. VERIFY Egress_Time ~= (T2-T1) +/- **Internal Processing Fail Time**

[Test 7.3.1.X41.Y3 is already correct. Not changing here.]

**** Note this test was already corrected via one of the Addenda for TP 20.0.1. No changes made here.

BTL-20.0.1 Errata-3: Correct Tests 7.3.2.X56.7 [BTLWG-1301]

Overview:
Typo in Step 12:
12. VERIFY Access_Event = LOCKOUT_RELINQUISH

in Step 26 it would be correct:
26. VERIFY (Access_Event = **LOCKOUT_RELINQUISHED**)

Changes:

Checklist Changes

None

Test Plan Changes

Specified Test Changes

[Change Test 7.3.2.X56.7]
from:
11. VERIFY Access_Event = LOCKOUT_RELINQUISH
to
11. VERIFY Access_Event = LOCKOUT_RELINQUISHED

BTL-20.0.1 Errata-4: Correct Test Reference in Test Plan Section 3.17.4 [BTLWG-1302]

Overview:
test 7.3.2.21.3.9 Read-only Recipient_List for external Notification Forwarder Objects is defined in 135.1-2019.
but in Test Plan 20.0 and 20.0.1 in chapter 3.17.4 it is referenced without "3":

135.1-2019 - 7.3.2.21.9 Read-only Recipient_List for external Notification Forwarder Objects

Changes:

Checklist Changes

None

Test Plan Changes

[Modify references for Test Read-only Recipient_List for external NotificationForwarder Objects in section 3.17.4]

135.1-2019 - 7.3.2.21.3.9 Read-only Recipient_List for external Notification Forwarder Objects		
	Test Conditionality	Must be executed if the IUT does not claim support for Notification Forwarder objects.
	Test Directives	
	Testing Hints	

Specified Test Changes

[None]

BTL-20.0.1 Errata-5: Correct Test 8.5.17.X1.2 [BTLWG-1318]

Overview:
In BTL Specified Tests 20.0.1-final Test 8.5.17.X1.2 step 1:
currently:
... pCurrentReliability = FAULT_LISTED
should be:
... pCurrentReliability = FAULTS_LISTED

Changes:

Checklist Changes

None

Test Plan Changes

None

Specified Test Changes

[Modify Test 8.6.17.X1.2]
from:
1. VERIFY pCurrentReliability = FAULT_LISTED
to
1. VERIFY pCurrentReliability = FAULTS_LISTED

BTL-20.0.1 Errata-6: Correct Tests 7.3.1.22.1 and 7.3.1.22.2 [BTLWG-1219]

Overview:
Test 7.3.1.X9.1 was incorrectly updated to the 135.1-2019 version when the Test Package was updated from 135.1-2013 to 135.1-2019. The update effectively dropped the change resulting from CR-0417.

This proposal takes the TP16 version of the test and recasts it as changes to the 135.1-2019 tests.

In addition, a couple of missed reference updates are made in the Access Control section.

V1
- Initial draft

Proposed Changes:

Checklist Changes

[None]

Test Plan Changes

[Replace all references to BTL - 7.3.1.X9.1 with BTL - 7.3.1.22.1]
[Replace all references to BTL - 7.3.1.X9.2 with 135.1-2019 - 7.3.1.22.2]
[Replace all references to 135.1-2019 - 7.3.1.22.1 with BTL - 7.3.1.22.1]

Specified Test Changes

[Modify 7.3.1.22.1 from 135.1-2019 and move into BTL Specified Tests]

7.3.1.22.1 Event_Detection_Enable Inhibits Event Generation
Reason for Change: remove expectation that Event_Detection_Enable is able to be equal to both TRUE and FALSE as per CR-0417.

Purpose: To verify that Event_Detection_Enable ~~enables and~~ disables event detection ~~in objects which are configured for event reporting~~

Test Concept: Select an event generating object, O1, that ~~is configured for~~ *supports* event reporting. ~~If possible, make the object generate an event, to an offnormal, so that if the object can have a non-normal state, it enters that state early in the test. This will help detect incorrect implementations that initiate a TO_NORMAL transition when the algorithm is disabled. Set the Event_Detection_Enable property to FALSE.~~ Verify the Event_State is NORMAL and ~~the~~ Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts are equal to their respective initial conditions, as mandated in the standard. ~~Repeat the process that made the object generate an event and observe that no notification messages are transmitted.~~ *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).*~~to detect and report unconfirmed events and requires acknowledgments for all transitions. Event_Detection_Enable is equal to TRUE. D1 is either the pTimeDelay, or pTimeDelayNormal parameter, or 0 (for transitions to and from FAULT state) depending on the event transition.~~ 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

~~1. VERIFY Event_Detection_Enable = TRUE~~

~~2. MAKE (a condition exist which will cause O1 to transition, to an offnormal state if possible)~~

~~3. WAIT D1~~

~~4. BEFORE **Notification Fail Time**~~

~~RECEIVE UnconfirmedEventNotification-Request~~

~~'Process Identifier' = (any valid process identifier),~~

~~'Initiating Device Identifier' = IUT,~~

~~'Event Object Identifier' = O1,~~

~~'Time Stamp' = (any valid time stamp),~~

~~'Notification Class' = (the notification class configured for O1),~~

~~'Priority' = (the value configured for the transition),~~

~~'Event Type' = (any valid event type),~~

~~'Message Text' = (optional, any valid message text),~~

~~'Notify Type' = (value from the Notify_Type property configured for O1),~~

~~'AckRequired' = TRUE,~~

~~'From State' = (any valid event state),~~

~~'To State' = (any event state appropriate to the event type),~~

~~'Event Values' = (any values appropriate to the event type)~~

~~5. IF Event_Detection_Enable is writable THEN~~

~~WRITE Event_Detection_Enable = FALSE~~

~~ELSE~~

~~MAKE (Event_Detection_Enable to FALSE. This property is expected to be set during system configuration and is not expected to change dynamically.)~~

~~6. WAIT (D1 + **Notification Fail Time + Internal Processing Fail Time)**~~

~~7. CHECK (that the IUT did not send any further event notifications for O1)~~

~~8~~*2*. VERIFY pCurrentState = NORMAL

~~9~~*3*. VERIFY Acked_Transitions = (T,T,T)

~~10~~*4*.~~IF (Protocol_Revision is present AND Protocol_Revision ≥ 1) THEN~~

VERIFY Event_Time_Stamps = [NO_TS, NO_TS, NO_TS]

~~11~~*5*.IF (Event_Message_Texts property exists) THEN

VERIFY Event_Message_Texts = ["", "", ""]

~~12~~*6*.MAKE (a condition exist which would cause O1 to transition, if Event_Detection_Enable were TRUE)

~~13~~*7*.WAIT (~~D1~~*DELAY* + **Notification Fail Time)**

~~14~~*8*.CHECK (that the IUT did not send any event notifications for O1)

~~15~~*9*.VERIFY pCurrentState = NORMAL

~~16~~*10*. VERIFY Acked_Transitions = (T,T,T)

~~17~~*11*. ~~IF (Protocol_Revision is present AND Protocol_Revision ≥ 1) THEN~~

VERIFY Event_Time_Stamps = [NO_TS, NO_TS, NO_TS]

~~18~~*12*. IF (Event_Message_Texts property exists) THEN

VERIFY Event_Message_Texts = ["", "", ""]

Notes to Tester: This behavior can alternately be tested using the ConfirmedEventNotification service, but it is not necessary to test both.

BTL-20.0.1 Errata-7: Fix Typos in Test 9.1.2.1 and 9.1.2.5 [BTLWG-1327, BTLWG-1346, BTLWG-1349]

Overview:

[BTLWG-1327]
In TP 20.0.1 final SPECIFIED TESTS EventNotificationRequests in a couple of places the argument 'Message Text' is written as 'Message Type':
Test 9.1.2.1:
Step 3,
Step 5,
Step 13 (two times),
Step 15 (in ELSE-branch)

AND

[BTLWG-1346]
Test 9.1.2.1 has in title and purpose something like “unsuccessful ACK with timestamp too old”
But in step 8:

8. TRANSMIT AcknowledgeAlarm-Request,
    'Acknowledging Process Identifier' = (the value of the 'Process Identifier' parameter in the event notification),
    'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),
    'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),
    'Time Stamp' = (any valid time stamp),
    'Acknowledgement Source' = (any valid value),
    'Time of Acknowledgment' = (the current time using a Time format)

the specification for ‘Time Stamp’ is “any valid time stamp” allowing the correct timestamp for the ACK or a newer timestamp valid to be issued which should not result in the expected error code.

Change the specification of 'Time Stamp; to:

'Time Stamp' = (any valid time stamp older than T1),

In addition, step 11 should be improved to use “T1”:

11. TRANSMIT AcknowledgeAlarm-Request,
    'Acknowledging Process Identifier' = (the process identifier configured for this event),
    'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),
    'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),
    'Time Stamp' = T1,
    'Time of Acknowledgment' = (the current time using a Time format)

AND

[BTLWG-1349]
In Specified Tests 20.0.1 test 9.1.2.1 was changed so instead of only normal -> non-normal transitions it can deal with arbitrary transitions
Step 7 was changed to handle this change however Step 10 was missed.

The same problem exists in test 9.1.2.5 in step 8.

Checklist Changes

None

Test Plan Changes

None

Specified Test Changes

[Replace all occurrences of 'Message Type' with 'Message Text' in test 9.1.2.1 in BTL Specified Tests. This is a BTL Specified Test change only, the 135.1 version of the test has this correct.]

[Fix up Timestamp references in step 8 and 11 per notes above.]

9.1.2.1 Unsuccessful Alarm Acknowledgment of Confirmed Event Notifications Because the 'Time Stamp' is Too Old

Reason For Change: Made changes to include not supported transitions.

Purpose: To verify that an alarm remains unacknowledged if the time stamp in the acknowledgment does not match the most recent transition to the current alarm state.

Test Concept: An alarm is triggered that causes the IUT to notify the TD and one other device. The TD acknowledges the alarm using an old time stamp and verifies that the acknowledgment is not accepted by the IUT and that the IUT does not notify other devices that the alarm was acknowledged. The TD then acknowledges the alarm using the proper time stamp and verifies that the acknowledgment is properly noted by the IUT. The IUT notifies all other recipients that the alarm was acknowledged.

Configuration Requirements: The IUT shall be configured with at least one object that can detect alarm conditions and send confirmed notifications. The Acked_Transitions property shall have the value B'111' indicating that all transitions have been acknowledged. The TD and one other BACnet device, if the IUT supports multiple recipients, shall be recipients of the alarm notification. *D1 is either the pTimeDelay, or pTimeDelayNormal parameter, or 0 (for transitions to and from FAULT state) depending on the event transition.*

Notes to Tester: The destination address used for the acknowledgment notification in step 11 shall be the same address used in step 3. If the IUT can only be configured with one recipient in the Recipient_List property of the issuing Notification_class object, omit steps 5, 6, 15, and 16.

Test Steps:

1. MAKE (a change that triggers the detection of an alarm event in the IUT)
2. WAIT (~~pTimeDelay~~*D1*)
3. BEFORE Notification Fail Time
    RECEIVE ConfirmedEventNotification-Request,
        'Process Identifier' = (the process identifier configured for this event),
        'Initiating Device Identifier' = IUT,
        'Event Object Identifier' = (the object detecting the alarm),
        'Time Stamp' = (T1: any valid time stamp),
        'Notification Class' = (the notification class configured for this event),
        'Priority' = (the priority configured for this event type),
        'Event Type' = (E1: any valid event type),
        'Message Text' = (optional, any valid message text),
        'Notify Type' = (the notify type configured for the event),
        'AckRequired' = TRUE,
        'From State' = ~~NORMAL~~*(any appropriate event state)*,
        'To State' = (S1: any appropriate ~~non-normal~~ event state),
        'Event Values' = (the values appropriate to the event type)
4. TRANSMIT BACnet-SimpleACK-PDU
5. RECEIVE
    DESTINATION = (a device other than the TD),
    SOURCE = IUT,

ConfirmedEventNotification-Request,
'Process Identifier' = (the process identifier configured for this event),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the object detecting the alarm),
'Time Stamp' = (T1),
'Notification Class' = (the notification class configured for this event),
'Priority' = (the priority configured for this event type),
'Event Type' = (E2: any valid event type),
'Message Text' = (optional, any valid message text),
'Notify Type' = (the notify type configured for the event),
'AckRequired' = TRUE,
'From State' = ~~NORMAL~~*(any appropriate event state)*,
'To State' = (S2: any appropriate ~~non-normal~~ event state),
'Event Values' = (the values appropriate to the event type)

6. TRANSMIT BACnet-SimpleACK-PDU
7. VERIFY (the 'Event Object Identifier' from the event notification), Acked_Transitions = ~~(FALSE,TRUE,TRUE)~~*(appropriate bit FALSE, the others TRUE)*
8. TRANSMIT AcknowledgeAlarm-Request,
   'Acknowledging Process Identifier' = (the value of the 'Process Identifier' parameter in the event notification),
   'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),
   'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),
   'Time Stamp' = (any valid time stamp *older than T1*),
   'Acknowledgement Source' = (any valid value),
   'Time of Acknowledgment' = (the current time using a Time format)
9. RECEIVE BACnet-Error-PDU
   Error Class = SERVICES,
   Error Code = INVALID_TIME_STAMP
10. VERIFY (the 'Event Object Identifier' from the event notification), Acked_Transitions = ~~(FALSE,TRUE,TRUE)~~*(appropriate bit FALSE, the others TRUE)*
11. TRANSMIT AcknowledgeAlarm-Request,
   'Acknowledging Process Identifier' = (the process identifier configured for this event),
   'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),
   'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),
   'Time Stamp' = ~~(the time stamp conveyed in the notification)~~*T1*,
   'Time of Acknowledgment' = (the current time using a Time format)
12. RECEIVE BACnet-Simple-ACK-PDU
13. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
   BEFORE Notification Fail Time
   RECEIVE
   ConfirmedEventNotification-Request,
   'Process Identifier' = (the process identifier configured for this event),
   'Initiating Device Identifier' = IUT,
   'Event Object Identifier' = (the object detecting the alarm),
   'Time Stamp' = (T2: any valid time stamp),
   'Notification Class' = (the notification class configured for this event),
   'Priority' = (the priority configured for this event type),
   'Event Type' = (any valid event type),
   'Message Text' = (optional, any valid message text),
   'Notify Type' = ACK_NOTIFICATION,
   'To State' = (S1 or S2)

ELSE
   BEFORE Notification Fail Time
   RECEIVE
   ConfirmedEventNotification-Request,
   'Process Identifier' = (the process identifier configured for this event),
   'Initiating Device Identifier' = IUT,
   'Event Object Identifier' = (the object detecting the alarm),
   'Time Stamp' = (T2: any valid time stamp),
   'Notification Class' = (the notification class configured for this event),
   'Priority' = (the priority configured for this event type),

'Event Type' = (any valid event type),
'Message Text' = (optional, any valid message text),
'Notify Type' = ACK_NOTIFICATION

14. TRANSMIT BACnet-SimpleACK-PDU
15. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
    RECEIVE
        DESTINATION = (a device other than the TD),
        SOURCE = IUT,
        ConfirmedEventNotification-Request,
            'Process Identifier' = (the process identifier configured for this event),
            'Initiating Device Identifier' = IUT,
            'Event Object Identifier' = (the object detecting the alarm),
            'Time Stamp' = (T2),
            'Notification Class' = (the notification class configured for this event),
            'Priority' = (the priority configured for this event type),
            'Event Type' = (any valid event type),
            'Message Text' = (optional, any valid message text),
            'Notify Type' = ACK_NOTIFICATION,
            'To State' = (S1 or S2)
    ELSE
    RECEIVE
        DESTINATION = (a device other than the TD),
        SOURCE = IUT,
        ConfirmedEventNotification-Request,
            'Process Identifier' = (the process identifier configured for this event),
            'Initiating Device Identifier' = IUT,
            'Event Object Identifier' = (the object detecting the alarm),
            'Time Stamp' = (T2),
            'Notification Class' = (the notification class configured for this event),
            'Priority' = (the priority configured for this event type),
            'Event Type' = (any valid event type),
            'Message Text' = (optional, any valid message text),
            'Notify Type' = ACK_NOTIFICATION
16. TRANSMIT BACnet-SimpleACK-PDU
17. VERIFY (the 'Event Object Identifier' from the event notification), Acked_Transitions = (TRUE,TRUE,TRUE)

~~Notes to Tester: The destination address used for the acknowledgment notification in step 11 shall be the same address used in step 3. If the IUT can only be configured with one recipient in the Recipient_List property of the issuing Notification_class object, omit steps 5, 6, 15, and 16.~~

[Modify step 8 to VERIFY the correct expected Acked_Transitions, per BTLWG-1349]

9.1.2.5 Unsuccessful Alarm Acknowledgment of Unconfirmed Event Notifications Because the 'Time Stamp' is Too Old

Reason For Change: Made changes to include not supported transitions.

Purpose: To verify that an alarm remains unacknowledged if the time stamp in the acknowledgment does not match the most recent transition to the current alarm state.

Test Concept: An alarm is triggered that causes the IUT to notify the TD and one other device. The TD acknowledges the alarm using an old time stamp and verifies that the acknowledgment is not accepted by the IUT and that the IUT does not notify other devices that the alarm was acknowledged. The TD then acknowledges the alarm using the proper time stamp and verifies that the acknowledgment is properly noted by the IUT. The IUT notifies all other recipients that the alarm was acknowledged.

Configuration Requirements: The IUT shall be configured with at least one object that can detect alarm conditions and send unconfirmed notifications. The Acked_Transitions property shall have the value B'111' indicating that all transitions have been acknowledged. The TD and one other BACnet device, if the IUT supports multiple recipients, shall be recipients of the

alarm notification. *D1 is either the pTimeDelay, or pTimeDelayNormal parameter, or 0 (for transitions to and from FAULT state) depending on the event transition.*

Notes to Tester: The destination address used for the acknowledgment notification in step 11 shall be the same address used in step 3. The destination address used for the acknowledgment notification in step 12 shall be the same address used in step 4. When multiple event notifications are expected for a specific event, the order that the IUT transmits them in is irrelevant. If the IUT can only be configured with one recipient in the Recipient_List property of the issuing Notification_class object, omit steps 4 and 12.

Test Steps:

1. MAKE (a change that triggers the detection of an alarm event in the IUT)
2. WAIT ~~pTimeDelay~~*D1*
3. BEFORE Notification Fail Time
    RECEIVE UnconfirmedEventNotification-Request,
        'Process Identifier' = (the process identifier configured for this event),
        'Initiating Device Identifier' = IUT,
        'Event Object Identifier' = (the object detecting the alarm),
        'Time Stamp' = (T1: any valid time stamp),
        'Notification Class' = (the notification class configured for this event),
        'Priority' = (the priority configured for this event type),
        'Event Type' = (any valid event type),
        'Message Text' = (optional, any valid message text),
        'Notify Type' = (the notify type configured for the event),
        'AckRequired' = TRUE,
        'From State' = ~~NORMAL~~*(any appropriate event state)*,
        'To State' = (S1: any appropriate ~~non-normal~~ event state),
        'Event Values' = (the values appropriate to the event type)
4. IF (the notification in step 3 was not a broadcast) THEN
    RECEIVE
        DESTINATION = (at least one device other than the TD),
        SOURCE = IUT,
        UnconfirmedEventNotification-Request,
            'Process Identifier' = (the process identifier configured for this event),
            'Initiating Device Identifier' = IUT,
            'Event Object Identifier' = (the object detecting the alarm),
            'Time Stamp' = (T1),
            'Notification Class' = (the notification class configured for this event),
            'Priority' = (the priority configured for this event type),
            'Event Type' = (any valid event type),
            'Message Text' = (optional, any valid message text),
            'Notify Type' = (the notify type configured for the event),
            'AckRequired' = TRUE,
            'From State' = ~~NORMAL~~*(any appropriate event state)*,
            'To State' = (S2: any appropriate ~~non-normal~~ event state),
            'Event Values' = (the values appropriate to the event type)
5. VERIFY (the 'Event Object Identifier' from the event notification), Acked_Transitions = ~~(FALSE,TRUE,TRUE)~~*(appropriate bit FALSE, the others TRUE)*
6. TRANSMIT AcknowledgeAlarm-Request,
    'Acknowledging Process Identifier' = (the value of the 'Process Identifier' parameter in the event notification),
    'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),
    'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),
    'Time Stamp' = (a time stamp older than the one conveyed in the notification),
    'Acknowledgement Source' = (any valid value),
    'Time of Acknowledgment' = (the TD's current time using a Time format)
7. RECEIVE BACnet-Error-PDU
    Error Class = SERVICES,
    Error Code = INVALID_TIME_STAMP
8. VERIFY (the 'Event Object Identifier' from the event notification), Acked_Transitions = ~~(FALSE,TRUE,TRUE)~~ *(appropriate bit FALSE, the others TRUE)*

9. TRANSMIT AcknowledgeAlarm-Request,
    'Acknowledging Process Identifier' = (the process identifier configured for this event),
    'Event Object Identifier' = (the 'Event Object Identifier' from the event notification),
    'Event State Acknowledged' = (the state specified in the 'To State' parameter of the notification),
    'Time Stamp' = (the time stamp conveyed in the notification),
    'Acknowledgement Source' = (any valid value),
    'Time of Acknowledgment' = (the TD's current time using a Time format)
10. RECEIVE BACnet-Simple-ACK-PDU
11. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
    BEFORE **Notification Fail Time**
        RECEIVE
            DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST | TD,
            SOURCE = IUT,
            UnconfirmedEventNotification-Request,
                'Process Identifier' = (the process identifier configured for this event),
                'Initiating Device Identifier' = IUT,
                'Event Object Identifier' = (the object detecting the alarm),
                'Time Stamp' = (T2: any valid time stamp),
                'Notification Class' = (the notification class configured for this event),
                'Priority' = (the priority configured for this event type),
                'Event Type' = (any valid event type),
                'Message Text' = (optional, any valid message text),
                'Notify Type' = ACK_NOTIFICATION,
                'To State' = (S1 or S2)
    ELSE
    BEFORE **Notification Fail Time**
        RECEIVE
            DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST | TD,
            SOURCE = IUT,
            UnconfirmedEventNotification-Request,
            'Process Identifier' = (the process identifier configured for this event),
            'Initiating Device Identifier' = IUT,
            'Event Object Identifier' = (the object detecting the alarm),
            'Time Stamp' = (T2: any valid time stamp),
            'Notification Class' = (the notification class configured for this event),
            'Priority' = (the priority configured for this event type),
            'Event Type' = (any valid event type),
            'Message Text' = (optional, any valid message text),
            'Notify Type' = ACK_NOTIFICATION
12. IF (the notification in step 11 was not broadcast) THEN
    IF (Protocol_Revision is present AND Protocol_Revision ≥ 1) THEN
        RECEIVE
            DESTINATION = (at least one device other than the TD),
            SOURCE = IUT,
            UnconfirmedEventNotification-Request,
            'Process Identifier' = (the process identifier configured for this event),
            'Initiating Device Identifier' = IUT,
            'Event Object Identifier' = (the object detecting the alarm),
            'Time Stamp' = (T2),
            'Notification Class' = (the notification class configured for this event),
            'Priority' = (the priority configured for this event type),
            'Event Type' = (any valid event type),
            'Message Text' = (optional, any valid message text),
            'Notify Type' = ACK_NOTIFICATION,
            'To State' = (S1 or S2)
    ELSE
        RECEIVE
            DESTINATION = (at least one device other than the TD),
            SOURCE = IUT,
            UnconfirmedEventNotification-Request,

'Process Identifier' = (the process identifier configured for this event),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the object detecting the alarm),
'Time Stamp' = (T2),
'Notification Class' = (the notification class configured for this event),
'Priority' = (the priority configured for this event type),
'Event Type' = (any valid event type),
'Message Text' = (optional, any valid message text),
'Notify Type' = ACK_NOTIFICATION

13. VERIFY (the 'Event Object Identifier' from the event notification), Acked_Transitions = (TRUE,TRUE,TRUE)

~~Notes to Tester: The destination address used for the acknowledgment notification in step 11 shall be the same address used in step 3. The destination address used for the acknowledgment notification in step 12 shall be the same address used in step 4. When multiple event notifications are expected for a specific event, the order that the IUT transmits them in is irrelevant. If the IUT can only be configured with one recipient in the Recipient_List property of the issuing Notification_class object, omit steps 4 and 12.~~

BTL-20.0.1 Errata-8: Fix Test 9.9.1.2 [BTLWG-1345, BTLWG-1347]

Overview:

Test 9.9.1.2 in Specified Tests 20.0.1 has in Step 5 the line:

5. VERIFY Oi, Tracking_Value = Tvi

The problem is that this is the first mentioning of "Tvi" so there is nothing to "VERIFY" against.

The verify should be changed to mimic the verification done in step 3:

5. VERIFY Oi, Tracking_Value = (Tvi: a normal value)

AND

Test 9.9.1.2 has been changed to work on an arbitrary list of objects.
But step 4 still only refers to two objects:

4. MAKE (remove the non-normal conditions for both objects)

should be changed to mimic step 8

4. MAKE (remove the non-normal conditions for each selected life safety object)

Changes:

Checklist Changes

None

Test Plan Changes

None

Specified Test Changes

[In BTL Specified Tests, Fix step 5 as shown below]

9.9.1.2 Reset Multiple Object Execution Tests
~~**9.9.2 Reset Multiple Object Execution Tests**~~

Reason for Change: The test is generalized to work for any number of objects. The test is written to not rely on Event_State so it can be applied to objects which do not support event reporting.

~~BACnet Reference Clause: 13.13.~~

Purpose: To verify that the IUT *correctly resets multiple latched life safety objects when executing* ~~can correctly execute~~ a LifeSafetyOperation service request ~~to multiple Life Safety objects~~ *not directed at a specified life safety object*.

Test Concept: ~~Two~~ *Multiple* Life Safety objects, O1 … On, are toggled between a ~~NORMAL and an OFFNORMAL state~~ *normal and non-normal BACnetLifeSafetyState*. The ~~STATUS~~ *objects' Present_Value properties* should latch in the ~~OFFNORMAL~~ *non-normal* state until the LifeSafetyOperation service request is ~~transmitted~~ *executed*. *This is repeated for*

each LifeSafetyOperation request type supported by the device. ~~This test may be omitted when the device does not support latching.~~

Configuration Requirements: ~~The IUT must support a Life Safety Object that supports alarming and have at least two objects that can be latched. This object is configured into the NORMAL state, driven to the OFFNORMAL state *or FAULT* and returned to the NORMAL state by adjusting its Present_Value. The STATUS should remain latched in the OFFNORMAL state until the reset is issued. This test must be repeated for each reset function supported.~~ *The Life-Safety objects start the test with their Present_Value properties in a normal state as interpreted by the life safety objects. This test shall be skipped if the device does not support latching Life Safety objects. If the IUT supports a single latching life safety object, apply this test to the single object.*

Test Steps:

1. REPEAT X = (All supported enumerations that reset the *objects* ~~object~~) DO {
2. MAKE (the selected objects enter ~~a~~ latched *non-normal* ~~state~~ *states* where enumeration X will reset the objects)
3. *REPEAT Oi = (each selected life safety object) {*
   VERIFY Oi, Present_Value = (Si: a non-normal value)
   VERIFY Oi, Tracking_Value = Si
   }
4. MAKE (~~Event_State = NORMAL~~*remove the non-normal conditions for each selected life safety object*)
5. *REPEAT Oi = (each selected life safety object) {*
   VERIFY Oi, Present_Value = Si
   VERIFY Oi, Tracking_Value = (Tvi: a normal value)
   }
   ~~IF (enumeration value creates an Event_State = FAULT) THEN~~
   ~~CHECK (Event_State = FAULT)~~
   ~~ELSE~~
   ~~CHECK (Event_State = OFFNORMAL)~~
6. TRANSMIT LifeSafetyOperation-Request,
   'Requesting Process Identifier' = (any valid identifier),
   'Requesting Source' = (any valid character string),
   'Request' = *X,* ~~(any valid LifeSafetyOperation request),~~
7. RECEIVE BACnet-SimpleACK-PDU
~~7. VERIFY (Object), STATUS = NORMAL~~
8. *REPEAT Oi = (each selected life safety object) {*
   VERIFY Oi, Present_Value = Tvi
   }

}

BTL-20.0.1 Errata-9: Advanced Modify BIBBs Needed Directives Update for Test 8.22.5 [BTLWG-1351]

Overview:

Test Directives for 8.22.5 in all Advanced Modify BIBBs should be the same as in DS-AM-A:
Repeat the test for all priority values, 1 through 16.

Changes:

Checklist Changes

None

Test Plan Changes

[Section 4.14.1 Data Sharing - Advanced Modify - A Base Requirements has the correct directives for test 8.22.5, copy into the following sections:
Section 4.30.1 Data Sharing - Life Safety Advanced Modify - A Base Requirements
Section 4.34.1 Data Sharing - Access Control Advanced Modify A Base Requirements
Section 4.51.1 Data Sharing - Lighting Advanced Modify - A Base Requirements
]

Specified Test Changes

None.

BTL-20.0.1 Errata-10: Modify Test Plan Standard Reference [BTLWG-1359]

Overview:

Section 1.1 of the Test Plan 20.0.1 references 135-2016 and many addenda because 135-2020 includes PR 21 and 22. Test Plan 23 can now include the reference to 135-2020, and the one addendum cd which brings us to PR 23.

Changes:

Checklist Changes

None

Test Plan Changes

[Update Section 1.1 to reference ASHRAE standard 135-2020 and Addendum cd.]

Document name	Description
135.1-2019	SSPC Test Standard. This document is the same as 135.1-2013 with the addenda o, p, q and r.
135-2020	BACnet standard revision number 22. This document is the same as 135-2016 with the addenda *bd, be, bi, bj, bk, bl, bm, bn, bp, bq, br, bs, bt, bu, bw, by, and bz.*
~~135-2016~~	~~BACnet standard revision number 19.~~ ~~This document is the same as 135-2012 with the addenda *ai, aj, al, am, an, aq, ar, as, at, au, av, aw, ax, ay, az, ba, bc, bf, bg, bh, am, and bc.*~~
~~135-2016*bd*~~	~~Staging Object Type~~
~~135-2016*be*~~	~~Lighting BIBBs and Device Profiles~~
~~135-2016*bi*~~	~~Audit Reporting Object Type~~
~~135-2016*bk*~~	
~~135-2016*bl*~~	
~~135-2016*bm*~~	
~~135-2016*bn*~~	
~~135-2016*bp*~~	
~~135-2016 *bq*~~	
~~135-2016*bj*~~	~~BACnet Secure Connect Datalink~~
135-2020*cd*	TLS v1.3 Cipher Suite Application Profile for BACnet/SC. Approved August 31, 2021.

Specified Test Changes

None

BTL-20.0.1 Errata-11: Correct Test Reference in Elevator Bibb Section [BTLWG-1358]

Overview:

Found in the Interim Test Package but carried over into our Test Plan Addendum bs. Section 4.52.1 Base Requirements Test **135.1-2013** - 8.18.3 - Reading and Presenting Properties. Should not refer to 135.1-2013 but reference 135.1-2019.

Changes:

Checklist Changes

None

Test Plan Changes

[In BTL Test Plan, replace section 4.52 Data Sharing - Elevator View - A]

4.52.1 Base Requirements

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

~~135.1-2013~~*135.1-2019* - 8.18.3 - Reading and Presenting Properties		
	Test Conditionality	Must be executed if the IUT does not support DS-EAV-A.
	Test Directives	Repeat the test for each of the standard object types and associated properties specified by DS-EV-A.
	Testing Hints	

Specified Test Changes

None

BTL-20.0.1 Errata-12: Correct Typos in Elevator Bibb Section [BTLWG-1364]

Overview:

There are two instances of chapter 3.39.4, "5.39.4 Supports DM-OCD-A" and "5.39.4 Supports AE-AVM-A".

"5.39.4 Supports DM-OCD-A" but "Test Directives" are: "Verify that the IUT claims support for AE-VM-A".

"5.39.4 Supports DM-OCD-A" spelling error - "Elevatr" should be “Elevator”.

Changes:

Checklist Changes

None

Test Plan Changes

[In BTL Test Plan, replace section 3.39 Alarm and Event Management - Elevator Advanced View and Modify - A]

5.39.4 Supports DM-OCD-A

The IUT shall support DM-OCD-A in order to facilitate creation and deletion of ~~elevatr~~elevator.

Verify Checklist		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims support for ~~AE-AVM-A~~AE-OCD-A.
	Testing Hints	

~~5.39.4~~5.39.5 Supports AE-AVM-A

The IUT shall support AE-VM-A in order to facilitate configuration of alarm parameters by the user.

Verify Checklist		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims support for AE-AVM-A.
	Testing Hints	

Specified Test Changes

None

BTL-20.0.1 Errata-13: Fix Test 8.4.17.X9.15 [BTLWG-1362, BTLWG-1368]

Overview:

In Test 8.4.17.X9.15 in BTL Specified Tests 20.0.1, step 10 has an incorrect Priority.

In the same test, steps 4 and 10 should refer to Status_Flags property instead of using the parameter 'pStatusFlags'.

Changes:

Checklist Changes

None

Test Plan Changes

None

Specified Test Changes

In Test 8.4.17.X9.15 in BTL Specified Tests 20.0.1:

Step 10:

'Priority' = (the value configured to correspond to a TO_FAULT transition),

Should be:

TO_NORMAL vs TO_FAULT.

In Steps 4 and 10:

'Event Values' = (pCurrentReliability, pStatusFlags ...

Should be:

Status_Flags vs pStatusFlags.

BTL-20.0.1 Errata-14: Fix Test 8.4.X11 Event Values Reference [BTLWG-1361, BTLWG-1367]

Overview:

In BTL Specified Tests 20.0.1, Test 8.4.X11 refers to an incorrect Event_Values entry.

Changes:

Checklist Changes

None

Test Plan Changes

None

Specified Test Changes

In Test 8.4.X11 in BTL Specified Tests 20.0.1:

Steps 3 and 11 have:

'Event Values' = { pMonitoredValue, pStatusFlags, pAccessEventTag, pAccessEventType, pAccessCredential ...

It should be referencing 'pAccessEventTime'

BTL-20.0.1 Errata-13: Fix Reference in Network Forwarder Test [BTLWG-1363]

Overview:

In Test 7.3.2.30.7.3 in BTL Specified Tests 20.0.1, the reference is incorrect.

Changes:

Checklist Changes

None

Test Plan Changes

Modify reference 135.1-2019 - 7.3.2.30.7.3 to BTL - 7.3.2.30.7.3.

Specified Test Changes

Move test 7.3.2.30.7.3 from 135.1-2019 to BTL Specified Tests, the reference 7.3.2.31.2 should be 7.3.2.30.2 and the reference 7.3.2.31.3 should be 7.3.2.30.3.