BACnet® TESTING LABORATORIES ADDENDA

Addendum fix3 to BTL Test Package 26.0

Revision final
Revised 9/4/2025

Approved by the BTL Working Group on September 4, 2025;
Approved by the BTL Working Group Voting Members on September 23, 2025;
Published on September 24, 2025.

[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.

In the following document, language to be added to existing clauses within the BTL Test Package 26.0 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-26.0 fix3-1: Interpretation of Day-of-Week Field And Unspecified Value [BTLWG-1708, CR-0578]

Overview:

CR-0578 asked if it was a local matter if a device processed day-of-week fields in specified dates. A follow-up Interpretation Request was created and answered.

Interpretation: If a device receives a BACnet write request that contains a valid date with an invalid day of week, such as X'FF (the pattern used to denote an unspecified day of week), for a property that requires a specified date, it is a local matter whether the device accepts or rejects the request.

Question: Is this Interpretation correct?

Answer: Yes

Changes:

Checklist Changes

None

Test Plan Changes

[Change Clauses 3.29.4, 3.29.5, 5.14.5, 7.3.7, and 7.7.7]

3.29.4 Contains a Writable Present_Value Property, or Can be Placed Out_Of_Service

The IUT contains, or can be made to contain, a DateTime Value object that contains a writable Present_Value property.

~~135.1-2023~~*BTL* - 7.2.9 - DateTime Non-Pattern Properties Test		
	Test Conditionality	Must be executed if the IUT claims Protocol_Revision 11 or greater.
	Test Directives	Apply to the Present_Value property in a DateTime Value object.
	Testing Hints	
BTL - 9.23.2.21 - DateTime Non-Pattern Properties Test using WritePropertyMultiple Service		
	Test Conditionality	This test shall only be applied to devices claiming Protocol_Revision 11 or higher and which supports execution of WritePropertyMultiple.
	Test Directives	Apply to the Present_Value property in a DateTime Value object.
	Testing Hints	

3.29.5 Contains a Writable Relinquish_Default Property

The IUT contains, or can be made to contain, a DateTime Value object that contains a writable Relinquish_Default property.

~~135.1-2023~~*BTL* - 7.2.9 - DateTime Non-Pattern Properties Test		
	Test Conditionality	Must be executed if the IUT claims Protocol_Revision 11 or greater.
	Test Directives	Apply to the Relinquish_Default property in a DateTime Value object.
	Testing Hints	
BTL - 9.23.2.21 - DateTime Non-Pattern Properties Test using WritePropertyMultiple Service		
	Test Conditionality	This test shall only be applied to devices claiming Protocol_Revision 11 or higher and which supports execution of WritePropertyMultiple.
	Test Directives	Apply to the Relinquish_Default property in a DateTime Value object.
	Testing Hints	

5.14.5 Supports Start_Time and Stop_Time Properties

The IUT can be made to start and stop logging using these properties.

These properties are required to be present and writable in Event Log objects, if either is present.

135.1-2023 - 7.3.2.24.2 - Start_Time Test		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
135.1-2023 - 7.3.2.24.3 - Stop_Time Test		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
~~135.1-2023~~*BTL* - 7.2.9 - DateTime Non-Pattern Properties Test		
	Test Conditionality	Must be executed if the IUT claims Protocol_Revision 11 or greater.
	Test Directives	Apply to the Start_Time and again to the Stop_Time properties in an Event Log object.
	Testing Hints	
BTL - 9.23.2.21 - DateTime Non-Pattern Properties Test using WritePropertyMultiple Service		
	Test Conditionality	This test shall only be applied to devices claiming Protocol_Revision 11 or higher and which supports execution of WritePropertyMultiple.
	Test Directives	Apply to the Start_Time and again to the Stop_Time properties in an Event Log object.
	Testing Hints	

7.3.7 Supports Start_Time and Stop_Time Properties

The IUT can be made to start and stop logging using these properties.

These properties are required to be present and writable in trend log objects that are trending a BACnet property.

135.1-2023 - 7.3.2.24.2 - Start_Time Test		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
135.1-2023 - 7.3.2.24.3 - Stop_Time Test		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
~~135.1-2023~~*BTL* - 7.2.9 - DateTime Non-Pattern Properties Test		
	Test Conditionality	Must be executed if the IUT claims Protocol_Revision 11 or greater.
	Test Directives	Apply to the Start_Time and again to the Stop_Time properties in a Trend Log object.
	Testing Hints	
BTL - 9.23.2.21 - DateTime Non-Pattern Properties Test using WritePropertyMultiple Service		
	Test Conditionality	This test shall only be applied to devices claiming Protocol_Revision 11 or higher and which supports execution of WritePropertyMultiple.
	Test Directives	Apply to the Start_Time and again to the Stop_Time properties in a Trend Log object.
	Testing Hints	

7.7.7 Supports Start_Time and Stop_Time Properties

The IUT can be made to start and stop logging using these properties.

If present these properties are required to be writable.

135.1-2023 - 7.3.2.24.2 - Start_Time Test		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

135.1-2023 - 7.3.2.24.3 - Stop_Time Test		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
~~135.1-2023~~*BTL* - 7.2.9 - DateTime Non-Pattern Properties Test		
	Test Conditionality	Must be executed if the IUT claims Protocol_Revision 11 or greater.
	Test Directives	Apply to the Start_Time and again to the Stop_Time properties in a Trend Log *Multiple* object.
	Testing Hints	
BTL - 9.23.2.21 - DateTime Non-Pattern Properties Test using WritePropertyMultiple Service		
	Test Conditionality	This test shall only be applied to devices claiming Protocol_Revision 11 or higher and which supports execution of WritePropertyMultiple.
	Test Directives	Apply to the Start_Time and again to the Stop_Time properties in a Trend Log object.
	Testing Hints	

Specified Test Changes

[Move 135.1-2023 – 7.2.9 to BTL Specified Tests and modify]

7.2.9 DateTime Non-Pattern Properties Test

Reason For Change: Remove day of week test based on IR.

Purpose: To verify that the property being tested does not accept special date field values.

Test Concept: The property being tested, P1, is written with each of the special datetime field values to ensure that the property does not accept them. A datetime DT1 is selected which is within the range that the IUT will accept for the property. The value, V1, written to the property, is the datetime DT1 with one of its fields replaced with one of the date or time 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. *It is a local matter whether the device accepts or rejects an invalid day of week field so it is not tested.* ~~This test shall only be applied to devices claiming Protocol_Revision 11 or higher.~~

~~Notes to Tester: If P1 is an array, then an array index shall be provided in the TRANSMIT portion of step 1.~~

Test Steps:

1. REPEAT SV = (year unspecified, month unspecified, day of month unspecified, ~~day of week unspecified,~~ odd months, even months, last day of month, even days, odd days, hour unspecified, minute unspecified, second unspecified, hundredths unspecified) DO {
2. TRANSMIT WriteProperty-Request
   'Object Identifier' = O1,
   'Property Identifier' = P1,
   'Property Value' = (DT1 updated with the special value SV)
3. RECEIVE BACnet-Error-PDU
   'Error Class' = PROPERTY,
   'Error Code' = VALUE_OUT_OF_RANGE

}

[Change BTL Specified Tests - 9.23.2.21]

9.23.2.21 DateTime Non-Pattern Properties Test using WritePropertyMultiple Service

Reason for Change: Update Test Concept to include meaning of O1. *Remove day of week test based on IR.*

Purpose: To verify that the property being tested does not accept special date field values.

Test Concept: *O1 is the object being tested*. The property being tested, P_1, is written with each of the special datetime field values to ensure that the property does not accept them. A datetime DT_1 is selected which is within the range that the IUT will accept for the property. The value, V_1, written to the property is the datetime DT_1 with one of its fields replaced with one of the date or time 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. *It is a local matter whether the device accepts or rejects an invalid day of week field so it is not tested.*~~This test shall only be applied to devices claiming Protocol_Revision 11 or higher.~~

~~Notes to Tester: if P1 is an array, then a non-zero array index may be provided in the TRANSMIT and the same array index observed in the WritePropertyMultiple-Error.~~

Test Steps:

1. REPEAT SV = (year unspecified, month unspecified, day of month unspecified, ~~day of week unspecified,~~
   odd months, even months, last day of month, even days, odd days, hour unspecified, minute unspecified,
   second unspecified, hundredths unspecified) DO {
2. TRANSMIT WritePropertyMultiple-Request,
   'Object Identifier' = O1,
   'Property Identifier' = P1,
   'Property Value' = (DT_1 updated with the special value SV)
3. RECEIVE WritePropertyMultiple-Error,
   'Error Class' = PROPERTY,
   'Error Code' = VALUE_OUT_OF_RANGE,
   'Object Identifier' = Object1,
   'Property Identifier' = P1)
   | (BACnet-Reject-PDU
   'Reject Reason' = INVALID_PARAMETER_DATATYPE)
   | (BACnet-Reject-PDU
   'Reject Reason'= INVALID_TAG)
   }

BTL-26.0 fix3-2: Fix Original-Broadcast-NPDU Test [BTLWG-1712]

Overview:

Test 12.4.4.1.1 TD is not correctly defined.

Changes:

Checklist Changes

None

Test Plan Changes

[Move 135-2023 – 12.4.4.1.1 to BTL Specified Tests in Test Plan 9.8.4]

~~135.1-2023~~*BTL* - 12.4.4.1.1 - Original-Broadcast-NPDU

Specified Test Changes

[Move test 12.4.4.1.1 from 135.1-2023 into BTL Specified Tests, and modify]

12.4.4.1.1 Original-Broadcast-NPDU

Reason For Change: TD is not correctly defined.

Purpose: To verify that the IUT, configured as a BBMD, will forward an Original-Broadcast-NPDU request.

Configuration Requirements: The TD is a non-BBMD and on the same network as the IUT.

Test Steps:

1. TRANSMIT
   DA = B/IPv6 Link Local Multicast Address,
   SA = TD,
   Source-Virtual-Address = TD,
   Original-Broadcast-NPDU,
   Who-Is-Request
2. RECEIVE
   DA = BBMD1,
   SA = IUT,
   Forwarded-NPDU,
   Original-Source-Virtual-Address = TD
   Original-Source-B/IPv6-Address = TD
   Who-Is-Request
3. RECEIVE
   DA = BBMD2,
   SA = IUT,
   Forwarded-NPDU,
   Original-Source-Virtual-Address = TD
   Original-Source-B/IPv6-Address = TD
   Who-Is-Request
4. RECEIVE
   DA = BBMD3,

SA = IUT,
Forwarded-NPDU,
Original-Source-Virtual-Address = TD
Original-Source-B/IPv6-Address = TD
Who-Is-Request

BTL-26.0 fix3-3:DISCARD_CHANGES Error Code Fix [BTLWG-1714]

Overview:

The error code identified in 7.3.2.46.3.2.X2 DISCARD_CHANGES Command Failure Test does not align with the BACnet standard (see Clause 12.56.16).

Changes:

Checklist Changes

None

Test Plan Changes

None

Specified Test Changes

[In BTL Specified Tests, change]

7.3.2.46.3.2.X2 DISCARD_CHANGES Command Failure Test

Reason for change: No test existed. Incorrect error code.

Purpose: To verify that Network Port object responds to DISCARD_CHANGES commands when the command is not supported.

Test Concept: Attempt to command a Network Port which does not support the DISCARD_CHANGES. Verify that the attempt fails with an Error Class of PROPERTY and an error code of OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED.

Configuration Requirements: Select a Network Port which supports writable properties that set the Changes_Pending property to TRUE.

Test Steps:

1. TRANSMIT WriteProperty-Request,
   'Object Identifier' = (the Network Port object),
   'Property Identifier' = (any writable property that results in Changes_Pending = TRUE),
   'Property Value' = (any valid value)
2. RECEIVE BACnet-SimpleACK-PDU
3. TRANSMIT WriteProperty-Request,
   'Object Identifier' = (the Network Port object),
   'Property Identifier' = Command,
   'Property Value' = DISCARD_CHANGES,
4. RECEIVE BACnet-Error-PDU
   'Error Class' = PROPERTY,
   'Error Code' = OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED
5. VERIFY Command = IDLE

BTL-26.0 fix3-4: ReadPropertyMultiple Array Properties [BTLWG-1613]

Overview:

This issue is for BTLWG-1613 and BTL test '9.20.1.X2 ReadPropertyMultiple Array Properties'.

9.20.1.X2's step 1 is incorrect and its step 3 does not work if the tested array contains zero elements.

Changes:

Checklist Changes

None

Test Plan Changes

None

Specified Test Changes

[Modify test 9.20.1.X2 in BTL Specified Tests]

9.20.1.X2 ReadPropertyMultiple Array Properties
Reason for Change: No test exists for this functionality. This test is not in any SSPC proposal. *Correct test issue that occurred when array length was zero.*

Purpose: To verify that the IUT can execute ReadPropertyMultiple service requests when the requested property is an array~~, when its size as well as when a single element of the array is requested. Another request is made to read an element of an array where the array index is out of range~~.

Test Concept: The TD reads the size of the array property, and then reads the first and last entries in the array. Finally, the TD reads past the end of the array and ensures that the IUT returns the correct error.

Configuration Requirement: O1 is any object in the IUT database having array property P1.

Test Steps:

1. ~~VERIFY P1 = X,~~ READ X = (O1), P1, ARRAY INDEX = 0
2. IF (X>0) THEN
3. TRANSMIT ReadPropertyMultiple-Request,
   'Object Identifier' = O1,
   'Property Identifier' = P1,
   'Property Array Index' = 1
4. RECEIVE ReadPropertyMultiple-ACK,
   'Object Identifier' = O1,
   'Property Identifier' = P1,
   'Property Array Index' = 1,
   'Property Value' = (V, any valid value of the correct data type for property P1)
5. TRANSMIT ReadPropertyMultiple-Request,
   'Object Identifier' = O1,
   'Property Identifier' = P1,
   'Property Array Index' = X,
6. RECEIVE ReadPropertyMultiple-ACK,
   'Object Identifier' = O1,
   'Property Identifier' = P1,
   'Property Array Index' = X,

'Property Value' = (V, any valid value of the correct data type for property P1)

~~3. CHECK (V is any value of the correct data type for property P1)~~

47. TRANSMIT ReadPropertyMultiple-Request,
'Object Identifier' = O1,
'Property Identifier' = P1,
'Property Array Index' = (X+1)

~~5~~8. RECEIVE ReadPropertyMultiple-Error,
'Error Class' = PROPERTY,
'Error Code' = INVALID_ARRAY_INDEX
| ReadPropertyMultiple-ACK,
'Object Identifier' = O1,
'Property Identifier' = P1,
'Property Array Index' = X+1,
'Property Access Error' = (
'Error Class' = PROPERTY,
'Error Code' = INVALID_ARRAY_INDEX
)

BTL-26.0 fix3-5: Fix ValidDays Test to use Correct Time Delay [BTLWG-1649]

Overview:

7.3.2.21.3.1- ValidDays Test
In Step 2, is using the variable pTimeDelay in the WAIT statement. This is not correct.

The property pTimeDelay is a parameter for the event algorithm and is defined as follows in the ANSI/ASHRAE Standard 135-2020 in Chapter 13.3:

"This parameter, of type Unsigned, represents the time, in seconds, that the offnormal conditions must exist before an offnormal event state is indicated."

However, there is no transition to an OFFNORMAL state before Step 2. In fact, there is no transition between different states at all, only a TimeSync is taking place!

Changes:

Checklist Changes

None

Test Plan Changes

[Update all references to test 7.3.2.21.3.1 from 135.1-2023 to BTL]

Specified Test Changes

[Move test from 135.1-2023 to BTL Specified Tests and change as follows.]

7.3.2.21.3.1 ValidDays Test

Reason For Change: Fix delay reference.

Purpose: To verify the operation of the Valid Days parameter of a BACnetDestination as used in the Recipient_List property of the Notification Class object.

Test Concept: The TD will select one instance of the Notification Class object and one instance of an event-generating object that is linked to the Notification Class object. The Recipient_List of the Notification Class object shall contain a single recipient with the Valid Days parameter configured so that at least one day is TRUE and at least one day is FALSE. The properties of the event-generating object will be manipulated to cause the Event_State to change from NORMAL to OFFNORMAL. The tester verifies that if the local date is one of the valid days a notification message is transmitted and if the local date is not a valid day then no notification message is transmitted. For devices that implement a read-only Recipient_List property for all instances of Notification Class objects and are exclusively configured for all days (Valid Days set to all Days), this test shall be omitted. For devices that implement a writeable Recipient_List property for all instances of Notification Class objects, and exclusively accept all days as the only permitted configuration, this test shall be omitted.

Configuration Requirements: The IUT shall be configured with one or more instance of the Notification Class object and at least one event-generating object that is linked to the Notification Class object. The event-generating object may be any object that supports intrinsic reporting or it may be an Event Enrollment object. The event-generating object shall have the Event_Enable property configured to transmit notification messages for all event transitions. The event-generating object shall be configured to be in a NORMAL event state at the start of the test. The Notification Class object shall be configured with a single recipient in the Recipient_List. The Valid Days parameter shall be configured so that at least one day of the week has a value of TRUE and at least one day of the week has a value of FALSE. The Transitions parameter shall be configured for the recipient to receive notifications for all event transitions.

Test Steps:

1. (TRANSMIT TimeSynchronization-Request,
'Time' = (any time within the window defined by From Time and To Time in the
BACnetDestination that corresponds to one of the valid days)) |
(TRANSMIT UTCTimeSynchronization-Request,
'Time' = (any time within the window defined by From Time and To Time in the
BACnetDestination that corresponds to one of the valid days, converted
to UTC)) |
MAKE (the local date and time = (any time within the window defined by From Time and
To Time in the BACnetDestination that corresponds to one of the valid days))
2. ~~WAIT (pTimeDelay + Notification Fail Time)~~
WAIT (InternalProcessingFailTime)
3. VERIFY pCurrentState = NORMAL
4. IF (pMonitoredValue is writable) THEN
WRITE pMonitoredValue = (a value that is OFFNORMAL)
ELSE
MAKE (pMonitoredValue have a value that is OFFNORMAL)
5. WAIT (pTimeDelay)
6. BEFORE Notification Fail Time
RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the event-generating object configured for this
test),
'Time Stamp' = (any valid time stamp),
'Notification Class' = (the class corresponding to the object being tested),
'Priority' = (the value configured to correspond to a TO-OFFNORMAL
transition),
'Event Type' = (any valid event type),
'Message Text' = (optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = OFFNORMAL,
'Event Values' = (values appropriate to the event type)
7. TRANSMIT BACnet-SimpleACK-PDU
8. VERIFY pCurrentState = OFFNORMAL
9. (TRANSMIT TimeSynchronization-Request,
'Time' = (any time within the window defined by From Time and To time in the
BACnetDestination that corresponds to one of the invalid days)) |
(TRANSMIT UTCTimeSynchronization-Request,
Time' = (any time within the window defined by From Time and To Time in the
BACnetDestination that corresponds to one of the invalid days, converted to UTC)) |
MAKE (the local date and time = (any time within the window defined by From Time and
To Time in the BACnetDestination that corresponds to one of the invalid days))
10. IF (pMonitoredValue is writable) THEN
WRITE pMonitoredValue = (a value that is NORMAL)
ELSE
MAKE (pMonitoredValue have a value that is NORMAL)
11. WAIT (~~pTimeDelay~~*pTimeDelayNormal* + **Notification Fail Time**)
12. CHECK (verify that no notification message was transmitted)

BTL-26.0 fix3-6: Fix Change of Value Notification Test [BTLWG-1650]

Overview:

In ASHRAE Standard 135.1-2023 for Test 8.2.1 “ReportedPV” has no defined value in Test Step 8 and 11 when Out_Of_Service is writable.

Changes:

Checklist Changes

None

Test Plan Changes

[Change all references for test 8.2.1 from 135.1-2023 to BTL]

Specified Test Changes

8.2.1 Change of Value Notification for Changes to Present_Value in Objects with a COV_Increment
Reason for Change: Defined value for ReportedPV for later use in step 3

Purpose: To verify that the IUT can initiate ConfirmedCOVNotification service requests conveying a change of the Present_Value property in Numeric Objects.

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

Configuration Requirements: At the beginning of the test, the Out_Of_Service property shall have a value of FALSE. Select an object where Present_Value is not expected to change outside the tester's control by more than COV_Increment or which has a writable Out_Of_Service. In devices where the COV_Increment is always less than the minimal change that Present_Value can make, skip steps 8 through 10.

Notes to Tester: The IUT may initiate additional COVNotifications. The final COVNotification shall accurately reflect Present_Value and Status_Flags.

Test Steps:

REPEAT X = (one supported object of each type) DO {
1. TRANSMIT SubscribeCOV-Request,
   'Subscriber Process Identifier' = (any value > 0 chosen by the TD),
   'Monitored Object Identifier' = X,
   'Issue Confirmed Notifications' = TRUE, 'Lifetime' = L
2. RECEIVE BACnet-SimpleACK-PDU
3. BEFORE Notification Fail Time
4. RECEIVE ConfirmedCOVNotification-Request,
   'Subscriber Process Identifier' = (the same value used in step 1),
   'Initiating Device Identifier' = IUT,
   'Monitored Object Identifier' = X,
   'Time Remaining' = (any value appropriate for the Lifetime selected),
   'List of Values' = (*ReportedPV* = the initial Present_Value and initial Status_Flags)
5~~4~~. TRANSMIT BACnet-SimpleACK-PDU
6~~5~~. TRANSMIT ReadProperty-Request,
   'Object Identifier' = X,
   'Property Identifier' = COV_Increment

76. RECEIVE BACnet-ComplexACK-PDU,
'Object Identifier' = X,
'Property Identifier' = COV_Increment,
'Property Value' = (a value "increment" that will be used below)
87. IF (Out_Of_Service is writable) THEN
9. WRITE X, Out_Of_Service = TRUE
10. BEFORE Notification Fail Time
11. RECEIVE ConfirmedCOVNotification-Request,
'Subscriber Process Identifier' = (the same value used in step 1),
'Initiating Device Identifier' = IUT,
'Monitored Object Identifier' = X,
'Time Remaining' = (any value appropriate for the Lifetime selected),
'List of Values' = (ReportedPV = the current Present_Value, and new Status_Flags)
12. TRANSMIT BACnet-SimpleACK-PDU
138. IF (Present_Value is now writable) THEN
14. WRITE X, Present_Value = (any value that differs from ReportedPV by less than "increment")
ELSE
15. MAKE (Present_Value = any value that differs from ReportedPV by less than "increment")
169. WAIT Notification Fail Time
1710. CHECK (verify that no COV notification was transmitted)
1811. IF (Present_Value is now writable) THEN
19. WRITE X, Present_Value = (any value that differs from ReportedPV by an amount greater than "increment")
ELSE
20. MAKE (Present_Value = any value that differs from ReportedPV by an amount greater than "increment")
2112. BEFORE NotificationFailTime
22. RECEIVE ConfirmedCOVNotification-Request,
'Subscriber Process Identifier' = (the same value used in step 1),
'Initiating Device Identifier' = IUT,
'Monitored Object Identifier' = X,
'Time Remaining' = (any value appropriate for the Lifetime selected),
'List of Values' = (the new Present_Value and new Status_Flags)
2313. TRANSMIT BACnet-SimpleACK-PDU
2414. TRANSMIT SubscribeCOV-Request,
'Subscriber Process Identifier' = (the same value used in step 1),
'Monitored Object Identifier' = X
2515. RECEIVE BACnet-SimpleACK-PDU
2616. IF (Out_Of_Service is writable) THEN
27. WRITE X, Out_Of_Service = FALSE
}

BTL-26.0 fix3-7: Fix 8.4.1.14 UNSIGNED_RANGE Test [BTLWG-1653]

Overview:

Inconsistent or rather wrong property name for high and low limit in Step 23 of test 8.4.14 in ANSI/ASHRAE Standard 135.1-2023.

Changes:

Checklist Changes

None

Test Plan Changes

[Change all references to test 8.4.14 from 135.1 to BTL]

Specified Test Changes

8.4.14 UNSIGNED_RANGE Test (ConfirmedEventNotification Test)
Reason for change: wrong property name for high and low limit in Step 23 (Low_Limit and High_Limit instead of pLowLimit and pHighLimit.

Purpose: To verify the correct operation of the UNSIGNED_RANGE event algorithm.

Test Concept: This test is the same as 8.4.6, except that the Event_Type is UNSIGNED_RANGE instead of OUT_OF_RANGE, and there is no pDeadband. If pMonitoredValue is not under the tester's control in the IUT, then pHighLimit and/or pLowLimit are modified to generate event notifications. The object begins the test in a NORMAL state. pMonitoredValue is raised to a value that is above the high limit. After the time delay expires, the object should enter the HIGH_LIMIT state and transmit an event notification message. pMonitoredValue is lowered to a value that is below the high limit. After the time delay expires, the object should enter the NORMAL state and issue an event notification. The same process is repeated to test the low limit.

Configuration Requirements: If possible, the IUT shall be configured such that the Event_Enable property has a value of TRUE for the TO_OFFNORMAL and TO_NORMAL transitions. If possible, pLimitEnable shall have a value of TRUE for both HighLimit and LowLimit events. The 'Issue Confirmed Notifications' parameter in the Recipient_List of the configured Notification Class shall have a value of TRUE. The Recipient_List of the configured Notification Class shall contain the TD, thus ensuring that notifications are emitted. The event-generating objects shall be in a NORMAL state at the start of the test.

Notes to Tester: The time stamps indicated by "" can have a value that indicates an unspecified time or a time that precedes the timestamp of the first received notification.*

Test Steps:

1. VERIFY pCurrentState = NORMAL
2. IF (pMonitoredValue is writable) THEN
   WRITE pMonitoredValue = (a value x: (x > pHighLimit))
   ELSE
   MAKE (pMonitoredValue have a value x: (x > pHighLimit))
3. WAIT (pTimeDelay)
4. BEFORE Notification Fail Time
   RECEIVE ConfirmedEventNotification-Request,
   'Process Identifier' = (any valid process ID),
   'Initiating Device Identifier' = IUT,
   'Event Object Identifier' = (the object being tested),

'Time Stamp' = (Toffnormal: any valid time stamp),
'Notification Class' = (the configured notification class),
'Priority' = (the value configured to correspond to a TO_OFFNORMAL transition),
'Event Type' = UNSIGNED_RANGE,
'Message Text' = (optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = HIGH_LIMIT,
'Event Values' = pMonitoredValue, pStatusFlags, pHighLimit

5. TRANSMIT BACnet-SimpleACK-PDU
6. IF (Protocol_Revision is present AND Protocol_Revision >= 13)) THEN
   VERIFY Status_Flags = (TRUE, FALSE, ?, ?)
7. VERIFY pCurrentState = HIGH_LIMIT
8. IF (Protocol_Revision is present AND Protocol_Revision >= 1) THEN
   VERIFY Event_Time_Stamps = (Toffnormal, *, *)
9. IF (pMonitoredValue is writable) THEN
   WRITE pMonitoredValue = (a value x: (pLowLimit < x < pHighLimit))
   ELSE
   MAKE (pMonitoredValue have a value x: (pLowLimit < x < pHighLimit))
10. WAIT (pTimeDelayNormal)
11. BEFORE Notification Fail Time
    RECEIVE ConfirmedEventNotification-Request,
    'Process Identifier' = (any valid process ID),
    'Initiating Device Identifier' = IUT,
    'Event Object Identifier' = (the object being tested),
    'Time Stamp' = (Tnormal: any valid time stamp),
    'Notification Class' = (the configured notification class),
    'Priority' = (the value configured to correspond to a TO_NORMAL transition),
    'Event Type' = UNSIGNED_RANGE,
    'Message Text' = (optional, any valid message text),
    'Notify Type' = EVENT | ALARM,
    'AckRequired' = TRUE | FALSE,
    'From State' = HIGH_LIMIT,
    'To State' = NORMAL,
    'Event Values' = pMonitoredValue, pStatusFlags, pHighLimit
12. TRANSMIT BACnet-SimpleACK-PDU
13. IF (Protocol_Revision is present AND Protocol_Revision >= 13)) THEN
    VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
14. VERIFY pCurrentState = NORMAL
15. IF (Protocol_Revision is present AND Protocol_Revision >= 1) THEN
    VERIFY Event_Time_Stamps = (Toffnormal, *, Tnormal)
16. IF (pMonitoredValue is writable) THEN
    WRITE pMonitoredValue = (a value x: (x < pLowLimit))
    ELSE
    MAKE (pMonitoredValue have a value x: (x < pLowLimit))
17. WAIT (pTimeDelay)
18. BEFORE Notification Fail Time
    RECEIVE ConfirmedEventNotification-Request,
    'Process Identifier' = (any valid process ID),
    'Initiating Device Identifier' = IUT,
    'Event Object Identifier' = (the object being tested),
    'Time Stamp' = (Tlowlimit: any valid time stamp),
    'Notification Class' = (the configured notification class),
    'Priority' = (the value configured to correspond to a TO_OFFNORMAL transition),
    'Event Type' = UNSIGNED_RANGE,
    'Message Text' = (optional, any valid message text),
    'Notify Type' = EVENT | ALARM,
    'AckRequired' = TRUE | FALSE,
    'From State' = NORMAL,

'To State' = LOW_LIMIT,
'Event Values' = pMonitoredValue, pStatusFlags, pLowLimit

19. TRANSMIT BACnet-SimpleACK-PDU
20. IF (Protocol_Revision is present AND Protocol_Revision >= 13)) THEN
    VERIFY Status_Flags = (TRUE, FALSE, ?, ?)
21. VERIFY pCurrentState = LOW_LIMIT
22. IF (Protocol_Revision is present AND Protocol_Revision >= 1) THEN
    VERIFY Event_Time_Stamps = (Tlowlimit, *, Tnormal)
23. IF (pMonitoredValue is writable) THEN
    WRITE pMonitoredValue = (a value x: ~~(Low_Limit < x < High_Limit)~~ *(pLowLimit < x < pHighLimit)*)
    ELSE
    MAKE (pMonitoredValue have a value x: ~~(Low_Limit < x < High_Limit)~~ *(pLowLimit < x < pHighLimit)*)
24. WAIT (pTimeDelayNormal)
25. BEFORE Notification Fail Time
    RECEIVE ConfirmedEventNotification-Request,
    'Process Identifier' = (any valid process ID),
    'Initiating Device Identifier' = IUT,
    'Event Object Identifier' = (the object being tested),
    'Time Stamp' = (Tlowtonormal: any valid time stamp),
    'Notification Class' = (the configured notification class),
    'Priority' = (the value configured to correspond to a TO_NORMAL transition),
    'Event Type' = UNSIGNED_RANGE,
    'Message Text' = (optional, any valid message text),
    'Notify Type' = EVENT | ALARM,
    'AckRequired' = TRUE | FALSE,
    'From State' = LOW_LIMIT,
    'To State' = NORMAL,
    'Event Values' = pMonitoredValue, pStatusFlags, pLowLimit
26. TRANSMIT BACnet-SimpleACK-PDU
27. IF (Protocol_Revision is present AND Protocol_Revision >= 13)) THEN
    VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
28. VERIFY pCurrentState = NORMAL
29. IF (Protocol_Revision is present AND Protocol_Revision >= 1) THEN
    VERIFY Event_Time_Stamps = (Tlowlimit, *, Tlowtonormal)

~~Notes to Tester: The time stamps indicated by "*" can have a value that indicates an unspecified time or a time that~~

BTL-26.0 fix3-8: Forcing Timer Expiration by Writing IDLE [BTLWG-1654]

Overview:

ANSI/ASHRAE Standard 135.1-2023:
7.3.2.47.1.12 Forcing Timer Expiration by Writing IDLE:

Step 8, test asks to verify the exact “Update_Time”. Since it is not possible to check the exact time for this step, it is suggested to replace " = " sign with "~=".

This issue also applies to the test "7.3.2.47.1.13 Resetting Timer by Writing IDLE", step 7.

Changes:

Checklist Changes

None

Test Plan Changes

[Change all references for test 7.3.2.47.1.12 from 135.1-2023 to BTL]

Specified Test Changes

7.3.2.47.1.12 Forcing Timer Expiration by Writing IDLE

Reason for Change: In Step 7 verifying the excact Update_Time against the current date and time is not possible.

Purpose: Interrupting the Timer while it is RUNNING, via a value of IDLE written to the Timer_State property.

Test Concept: Configure and start the Timer T1 to operate according to its values. Then write IDLE to Timer_State and observe that specified properties take their required values and all configured State_Change_Values transitions if any, take place.

Configuration Requirements: T1 starts this test with the Timer_State equal to RUNNING.

Test Steps:
1. VERIFY Timer_Running = TRUE
2. VERIFY Timer_State = RUNNING
3. WRITE Timer_State = IDLE
4. CHECK (IUT exhibits any changes configured in RUNNING_TO_IDLE transition)
5. VERIFY Timer_State = IDLE
6. VERIFY Last_State_Change = RUNNING_TO_IDLE
7. IF (Expiration_Time property is present in T1) THEN
   VERIFY Expiration_Time = (the unspecified datetime value)
8. IF (Update_Time property is present in T1) THEN
   VERIFY Update_Time ~= (the current date and time)
9. VERIFY Present_Value = 0

BTL-26.0 fix3-9: Fix 12.4.5.1 Execute Register-Foreign-Device [BTLWG-1656]

Overview:

In Step 3 the address of FD2 is requested but there is no FD2 in this test.

Changes:

Checklist Changes

None

Test Plan Changes

[Change all references to test 12.4.5.1 from 135.1 to BTL]

Specified Test Changes

[Move test 12.4.5.1 from 135.1 to BTL]

12.4.5.1 Execute Register-Foreign-Device

Reason for change: Step 3 requested the wrong address of a non-existing device.

Purpose: To verify that the IUT will handle a Register-Foreign-Device request.

Test Steps:

1. TRANSMIT
   DA = IUT,
   SA = TD,
   Source-Virtual-Address = TD,
   Register-Foreign-Device,
   'Time-To-Live' = 60
2. RECEIVE
   DA = TD,
   SA = IUT,
   Source-Virtual-Address = IUT,
   BVLC-Result,
   'Result Code' = 0
3. VERIFY NP, BBMD_Foreign_Device_Table = ((B/IPv6 address of ~~FD2~~ *TD*, 60, 90-execution time))

BTL-26.0 fix3-10: Fix Test 7.3.2.30.13.1 Recipient_List Persistence Test [BTLWG-1673]

Overview:

Separated Test Steps should be included in IF.

A restart should be started only if “IUT supports the ReinitializeDevice service” but the remaining steps are NOT in the “IF” (the “}” closes at the end of the same step).

If the IUT does not support the ReinitializeDevice service, then it has not restarted and the next steps cannot be performed.

Changes:

Checklist Changes

None

Test Plan Changes

[Change references for test 7.3.2.30.13.1 from 135.1 to BTL]

Specified Test Changes

7.3.2.30.13.1 Recipient_List Persistence Test

Reason for Change: Test Step 3,4 and 5 should be part of the IF in Test Step 2.

Purpose: This test insures that Recipient_List property value is maintained through a device “restart”.

Test Concept: Initialize the Recipient_List property with a known value, then cycle power or restart the IUT and verify the Recipient_List property value is maintained.

Configuration Requirements: Base setup 1 for Notification Forwarder object tests.

Test Steps:

1. MAKE (Recipient_List = {(all), -- Valid Days
(all), -- From Time, To Time
DEST_OBJ_ID, -- Recipient D1
DEST_PROCESS_ID, -- Any Process Identifier
DEST_CONF_NOTIF, -- Any Issue Confirmed Notifications
{T, T, T} -- Transitions
}) -- One list element
2. IF (IUT supports the ReinitializeDevice service) THEN {
3. TRANSMIT ReinitializeDevice-Request,
'Reinitialized State of Device' = WARMSTART,
'Password' = (any valid password) --if required by IUT
4. RECEIVE BACnet-SimpleACK-PDU
~~}~~
5~~3~~. CHECK (Did the IUT perform a WARMSTART restart)
6~~4~~. WAIT for restart to complete
7~~5~~. VERIFY Recipient_List = {(all), -- Valid Days
(all), -- From Time, To Time

```
                    DEST_OBJ_ID,                -- Recipient D1
                    DEST_PROCESS_ID,            -- Process Identifier
                    DEST_CONF_NOTIF,            -- Issue Confirmed Notifications
                    {T, T, T}                   -- Transitions
                     }                          -- One list element
```

}

8.~~6.~~ MAKE (The IUT reset by cycling power)

9.~~7.~~ WAIT for restart to complete

10.~~8.~~ VERIFY Recipient_List = {(all),

```
                    (all),                      -- Valid Days
                                                -- From Time, To Time
                   DEST_OBJ_ID,                 -- Recipient D1
                   DEST_PROCESS_ID,             -- Process Identifier
                   DEST_CONF_NOTIF,             -- Issue Confirmed Notifications
                   {T, T, T}                    -- Transitions
                   }                            -- One list element
```

BTL-26.0 fix3-11: Fix Subscribed_Recipients Persistence Test [BTLWG-1674]

Overview:

Separated Test Steps should be included in IF.

A restart should be started only if “IUT supports the ReinitializeDevice service” but the remaining steps are NOT in the “IF” (the “}” closes at the end of the same step)

If the IUT does not support the ReinitializeDevice service, then it has not restarted and the next steps cannot be performed.

Changes:

Checklist Changes

None

Test Plan Changes

[Change all references for test 7.3.2.30.13.2 from 135.1 to BTL]

Specified Test Changes

[Move test 7.3.2.30.13.2 from 135.1 to BTL and change as follows.]

7.3.2.30.13.2 Subscribed_Recipients Persistence Test

Reason for Change: Test Step 2,3 and 4 should be part of the IF in Test Step 1.

Purpose: This test insures that Subscribed_Recipients property values are maintained through a device “restart”.

Test Concept: Initialize the Subscribed_Recipients property with a known value, then cycle power to restart the IUT and verify the Subscribed_Recipients property value is maintained.

Configuration Requirements: Base setup 2 for Notification Forwarder object tests with TR lifetime sufficient for this test.

Note To Tester: Start_Up_TimeA or Start_Up_TimeB is the time in minutes required to restart the IUT.

Test Steps:

1. IF (IUT supports the ReinitializeDevice service) THEN {
2. TRANSMIT ReinitializeDevice-Request,
'Reinitialized State of Device' = WARMSTART,
'Password' = (any valid password) --if required by IUT
3. RECEIVE BACnet-SimpleACK-PDU
~~}~~
~~4.2.~~ CHECK (Did the IUT perform a WARMSTART restart)
~~5.3.~~ WAIT for restart to complete, making a note of the time to restart as Start_Up_TimeA
~~6.4.~~ VERIFY Subscribed_Recipients =
{DEST_OBJ_ID, -- Recipient D1
DEST_PROCESS_ID, -- Process Identifier
DEST_CONF_NOTIF, -- Issue Confirmed Notifications
TR1 -- Time Remaining where (TR–Start_Up_TimeA-1) <= TR1 <= TR
} -- One list element
}

7.~~5.~~ MAKE (The IUT reset by cycling power)
8.~~6.~~ WAIT for restart to complete, making a note of the time to restart as Start_Up_TimeB
9.~~7.~~ VERIFY Subscribed_Recipients =

{DEST_OBJ_ID, -- Recipient D1
DEST_PROCESS_ID, -- Process Identifier
DEST_CONF_NOTIF, -- Issue Confirmed Notifications
TR1 -- Time Remaining where (TR–Start_Up_TimeA-Start_Up_TimeB-2) <= TR1 <= TR
} -- One list element

~~Note To Tester: Start_Up_TimeA or Start_Up_TimeB is the time in minutes required to restart the IUT.~~

BTL-26.0 fix3-12: AE Full Presentation Test Directive Changes [BTLWG-1684]

Overview:

The test directives for 9.4.6 and 9.5.2 (Full Presentation) require testing with the Extended event type which can include a virtually infinite number of possible values in the 'Event Values' portion of an event notification. This work items changes the test directives to limit the scope.

Changes:

Checklist Changes

None

Test Plan Changes

[In BTL Test Plan, change test directives in section 5.18.1 (AE-AVN-A)]

5.18.1 Base Requirements

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

135.1-2023 - 9.4.6 - ConfirmedEventNotification Full Presentation		
	Test Conditionality	Must be executed.
	Test Directives	Repeat the test for each of the standard event types, including EXTENDED if IUT claims Protocol_Revision 13 or higher, and each of the transitions defined for those event types. ~~For notifications using the CHOICE format of extended,the presentation must show all the fields which were in the notification.~~ *For notifications with event-values containing constructed data and of type CHOICE, execute the test once for each CHOICE. Where the event-values contain an ANY type, the tester should limit testing to primitive datatypes.* Execute at least once with a Message_Text 256 or more characters in length.
	Testing Hints	
135.1-2023 - 9.5.2 - UnconfirmedEventNotification Full Presentation		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	Repeat the test for each of the standard event types, including EXTENDED if IUT claims Protocol_Revision 13 or higher, and each of the transitions defined for those event types. ~~For notifications using the CHOICE format of EXTENDED, the presentation must show all the fields which were in the notification.~~ *For notifications with event-values containing constructed data and of type CHOICE, execute the test once for each CHOICE. Where the event-values contain an ANY type, the tester should limit testing to primitive datatypes.* Execute at least once with a Message_Text 256 or more characters in length.

Specified Test Changes

None

BTL-26.0 fix3-13: Add CONFIGURATION_ERROR to Testing of Staging Object [BTLWG-1672, CR-0574]

Overview:

Problem:
In step 13, the property Reliability is written with all values except NO_FAULT_DETECTED. In this context, the value CONFIGURATION_ERROR is also written. According to ASHRAE, this value leads to the following behavior:

If Reliability has the value CONFIGURATION_ERROR, then Present_Value shall be set to Min_Pres_Value and Present_Stage to 1. (ASHRAE 12.62.6 page 614)

If the PV is set to Min_Pres_Value and the Present_Stage to 1 no Writes to Target_References will occur when Out_Of_Service is set back to false (step 14). Checks in step 17 will fail.

Annotation:
All variants of CONFIGURATION_ERROR condition and mechanism are tested in 7.3.2.50.8, 7.3.2.50.14, 7.3.2.50.15, 7.3.2.50.16

Changes:

Checklist Changes

None

Test Plan Changes

Test moves from ASHRAE-135.1 to BTL.

~~135.1-2023~~*BTL* - 7.3.2.50.11 - Out_Of_Service, Status_Flags, and Reliability for Staging Object		
	Test Conditionality	Must be executed
	Test Directives	
	Testing Hints	

Specified Test Changes

[Move test from ASHRAE-135.1 to BTL and modify.]

7.3.2.50.11 Out_Of_Service, Status_Flags, and Reliability for Staging Object
Reason for Change: CONFIGURATION_ERROR added in step 13, it was previously omitted.

Purpose: To verify that Present_Value and Reliability are writable when Out_Of_Service is TRUE, to verify the relationship between Out_Of_Service, Status_Flags, and Reliability, and to verify that writes to Target_References only occur when Out_Of_Service is FALSE.

Test Concept: The Out_Of_Service property is set to TRUE and the value of the Status_Flags property is validated. Present_Value is modified to verify that Present_Stage evaluates but writes to Target_References do not occur. If the IUT supports Reliability values other than NO_FAULT_DETECTED, writability for that property is tested and the value of the Status_Flags property is validated. The Out_Of_Service property is set to FALSE and the value of the Status_Flags property is validated. The Present_Value for one of the Target_References is checked to verify that it has the correct value, indicative of a write that occurred when transitioning Out_Of_Service from TRUE to FALSE.

Configuration Requirements: The Staging object used for this test shall be configured with at least one object in the Target_References property. The Stages property shall be configured with two stages such that Stages[S].Values = {V1…}

and Stages[S+1].Values = {V2…} where V1 <> V2. At the start of the test, the Staging object is properly configured such that Reliability = NO_FAULT_DETECTED and Present_Stage = S.

Test Steps:

1. READ SF1 = Status_Flags
2. VERIFY Reliability = NO_FAULT_DETECTED
3. VERIFY Present_Stage = S
4. READ O1 = Target_References, ARRAY INDEX = 1
5. VERIFY O1, Present_Value = V1
6. IF (Out_Of_Service is writable) THEN
   WRITE Out_Of_Service = TRUE
   ELSE
   MAKE (Out_Of_Service TRUE)
7. VERIFY Out_Of_Service = TRUE
8. VERIFY Status_Flags = (?, ?, ?, TRUE)
9. WRITE Present_Value = (PV: (Stages[S].Limit + Stages[S].Deadband) < PV < Stages[S+1].Limit)
10. VERIFY Present_Value = PV
11. VERIFY Present_Stage = S+1
12. VERIFY O1, Present_Value = V1
13. IF (the IUT supports Reliability values other than NO_FAULT_DETECTED) THEN
    REPEAT X = (all values of the Reliability enumeration appropriate to the object type except
    NO_FAULT_DETECTED *and CONFIGURATION_ERROR*) DO {
    WRITE Reliability = X
    VERIFY Reliability = X
    VERIFY Status_Flags = (?, TRUE, ?, TRUE)
    WRITE Reliability = NO_FAULT_DETECTED
    VERIFY Reliability = NO_FAULT_DETECTED
    VERIFY Status_Flags = (?, FALSE, ?, TRUE)
    }
14. IF (Out_Of_Service is writable) THEN
    WRITE Out_Of_Service = FALSE
    ELSE
    MAKE (Out_Of_Service FALSE)
15. VERIFY Status_Flags = SF1
16. VERIFY Reliability = NO_FAULT_DETECTED
17. IF (Present_Stage = S+1) THEN
    VERIFY O1, Present_Value = V2

BTL-26.0 fix3-14: 9.23.2.X - Writing first element of 'List of Write Access Specifications [BTLWG-1756]

Overview:

These tests cannot be executed/programmed correctly in its original form. You can't verify anything if you don't read it first.

Changes:

Checklist Changes

None

Test Plan Changes

[Modify all instances of tests 9.23.2.14, 9.23.2.15 and 9.23.2.17 from 135.2023 to BTL]

Specified Test Changes

9.23.2.14 Writing First Element of 'List of Write Access Specifications' with Object Access Error

Reason For Change: Replace VERIFY with READ.

Purpose: To verify the ability to correctly execute a WritePropertyMultiple service request for which the first element of the 'List of Write Access Specifications' contains a specification for an unsupported object and all writes after the first failed write attempt do not take place.

Test Concept: An attempt is made to write to a single property in two different objects. The first object is not supported. The second object is supported, and the property is writable. The objective is to verify that an appropriate error response is returned and that all writes after the first failed write attempt do not take place.

Configuration Requirements: If the IUT supports any writable scalar properties that are not commandable it shall be configured with one for use in this test. If no such properties are supported the IUT shall be configured with a writable array or commandable property and the test steps modified to account for this variation. In the test description O2 and P2 will be used to designate the writable object and property having value X used for this test. The designation ~~Bad Object~~ *BadObject* will be used to indicate an object that is not supported or not present in IUT databaseP1 is any valid Property Identifier.

Test Steps:

1. ~~VERIFY (O2), P2 = X~~ *READ X = (O2), P2*
2. TRANSMIT WritePropertyMultiple-Request,
   'Object Identifier' = BadObject,
   'Property Identifier' = P1,
   'Property Value' = (any valid value of the appropriate datatype for *P1*~~this property subject to the restrictions specified in the EPICS as defined in 4.4.2~~)
   'Object Identifier' = O2,
   'Property Identifier' = P2,
   'Property Value' = (any valid value not equal to X),
3. RECEIVE WritePropertyMultiple-Error,
   'Error Class' = OBJECT,
   'Error Code' = *(*UNKNOWN_OBJECT | *UNSUPPORTED_OBJECT_TYPE)*,
   'Object Identifier' = BadObject,
   'Property Identifier' = P1
   ~~'Property Identifier' = P1 |~~
   ~~(RECEIVE WritePropertyMultiple-Error,~~
   ~~'Error Class' = OBJECT,~~
   ~~'Error Code' = UNSUPPORTED_OBJECT_TYPE,~~
   ~~'Object Identifier' = BadObject,~~

~~'Property Identifier' = P1)~~
4. VERIFY (O2), P2 = X

9.23.2.15 Writing First Element of 'List of Write Access Specifications' with a Write Access Error

Reason For Change: Replace VERIFY with READ.

Purpose: To verify the ability to correctly execute a WritePropertyMultiple service request for which the first element of the 'List of Write Access Specifications' contains a specification for a read only property and all writes after the first failed write attempt do not take place.

Test Concept: An attempt is made to write to two properties in a single object. The first property is supported but read only. The second property is supported and writable. The objective is to verify that an appropriate error response is returned and that all writes after the first failed write attempt do not take place.

Configuration Requirements: If the IUT supports any writable scalar properties that are not commandable, it shall be configured with one for use in this test. If no such properties are supported, the IUT shall be configured with a writable array or commandable property and the test steps modified to account for this variation. In the test description, O1 will be used to designate the object, P1 the read only property having value X, P2 the writable property having value Y used for this test.

Test Steps:
1. ~~VERIFY (O1), P1= X~~ *READ X = (O1), P1*
2. ~~VERIFY (O1), P2=Y~~ *READ Y = (O1), P2*
3. TRANSMIT WritePropertyMultiple-Request,
   'Object Identifier' = O1,
   'Property Identifier' = ~~P1~~ *(P1, a read-only property in O1)*,
   'Property Value' = X,
   'Property Identifier' = P2,
   'Property Value' = (any valid value not equal to Y)
4. RECEIVE WritePropertyMultiple-Error,
   'Error Class' = PROPERTY,
   'Error Code' = WRITE_ACCESS_DENIED,
   'Object Identifier' = O1,
   'Property Identifier' = P1
5. ~~VERIFY (O1), P2 = Y~~ VERIFY (O1), P1 = X
6. VERIFY (O1), P2 = Y

9.23.2.17 Writing First Element of 'List of Write Access Specifications' with a Property Access Error

Reason For Change: Replace VERIFY with READ.

Purpose: To verify the ability to correctly execute a WritePropertyMultiple service request for which the first element of the 'List of Write Access Specifications' contains a specification for an unsupported property and all writes after the first failed write attempt do not take place.

Test Concept: An attempt is made to write to two properties in a single object. The first property is not supported for this object. The second property is supported for this object and writable. The objective is to verify that an appropriate error response is returned and that all writes after the first failed write attempt do not take place.

Configuration Requirements: If the IUT supports any writable scalar properties that are not commandable, it shall be configured with one for use in this test. If no such properties are supported, the IUT shall be configured with a writable array or commandable property and the test steps modified to account for this variation. In the test description, O1 will be used to designate the object, P1 the unsupported property, and P2 the writable property having value X used.

Test Steps:
1. ~~VERIFY (O1), P2 = X~~ *READ X = (O1), P2*
2. TRANSMIT WritePropertyMultiple-Request,
   'Object Identifier' = O1,
   'Property Identifier' = ~~P1~~ *(P1, an unsupported property for object O1)*,
   'Property Value' = (any valid value of the appropriate datatype for *P1*~~this property subject to the restrictions~~

~~specified in the EPICS as defined in 4.4.2~~)
'Property Identifier' = P2,
'Property Value' = (any valid value not equal to X),
3. RECEIVE WritePropertyMultiple-Error,
'Error Class' = PROPERTY,
'Error Code' = UNKNOWN_PROPERTY,
'Object Identifier' = O1,
'Property Identifier' = P1
4. VERIFY (O1), P2 = X

BTL-26.0 fix3-15: Exclude Presentation of Alarms Not Required by BIBBs [BTLWG-1764, CR-0594]

Overview:

Per CR-0594, the test directives for AE-VN-A (Simple Presentation) and AE-AVN-A (Full Presentation) incorrectly required presentation for all standard event types; However, the BIBB definition does not mandate support for CHANGE_OF_LIFE_SAFETY, ACCESS_EVENT, and BUFFER_READY event types.

Changes:

Checklist Changes

None

Test Plan Changes

[In BTL Test Plan, change test directives in sections 5.16.1 (AE-VN-A) and 5.18.1 (AE-AVN-A)]

5.16.1 Base Requirements

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

135.1-2023 - 9.4.5 - ConfirmedEventNotification Simple Presentation		
	Test Conditionality	Must be executed.
	Test Directives	Repeat the test for each of the standard event types, ~~including EXTENDED if IUT claims Protocol_Revision 13 or higher, and each of the transitions defined for those event types.~~ *excluding CHANGE_OF_LIFE_SAFETY, ACCESS_EVENT, and BUFFER_READY. If the IUT is Protocol_Revision 12 or lower, the EXTENDED event type shall also be excluded. For each event type tested, repeat the test for each of the transitions defined for that event type.* Execute at least once with a Message_Text 32 or more characters in length.
	Testing Hints	
135.1-2023 - 9.5.1 - UnconfirmedEventNotification Simple Presentation		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	Repeat the test for each of the standard event types, ~~including EXTENDED if IUT claims Protocol_Revision 13 or higher, and each of the transitions defined for those event types.~~ *excluding CHANGE_OF_LIFE_SAFETY, ACCESS_EVENT, and BUFFER_READY. If the IUT is Protocol_Revision 12 or lower, the EXTENDED event type shall also be excluded. For each event type tested, repeat the test for each of the transitions defined for that event type.* Execute at least once with a Message_Text 32 or more characters in length.

5.18.1 Base Requirements

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

135.1-2023 - 9.4.6 - ConfirmedEventNotification Full Presentation		
	Test Conditionality	Must be executed.

	Test Directives	Repeat the test for each of the standard event types, ~~including EXTENDED if IUT claims Protocol_Revision 13 or higher, and each of the transitions defined for those event types.~~ *excluding CHANGE OF LIFE SAFETY, ACCESS EVENT, and BUFFER_READY. If the IUT is Protocol_Revision 12 or lower, the EXTENDED event type shall also be excluded. For each event type tested, repeat the test for each of the transitions defined for that event type.* For notifications with event-values containing constructed data and of type CHOICE, execute the test once for each CHOICE. Where the event-values contain an ANY type, the tester should limit testing to primitive datatypes. Execute at least once with a Message_Text 256 or more characters in length.
	Testing Hints	
135.1-2023 - 9.5.2 - UnconfirmedEventNotification Full Presentation		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	Repeat the test for each of the standard event types, ~~including EXTENDED if IUT claims Protocol_Revision 13 or higher, and each of the transitions defined for those event types.~~ *excluding CHANGE OF LIFE SAFETY, ACCESS EVENT, and BUFFER_READY. If the IUT is Protocol_Revision 12 or lower, the EXTENDED event type shall also be excluded. For each event type tested, repeat the test for each of the transitions defined for that event type.* For notifications with event-values containing constructed data and of type CHOICE, execute the test once for each CHOICE. Where the event-values contain an ANY type, the tester should limit testing to primitive datatypes. Execute at least once with a Message_Text 256 or more characters in length.

Specified Test Changes

None

BTL-26.0 fix3-16: BSC Direct Connect - Invalid Certificate Test [BTLWG-1765, CR-0595]

Overview:

Remove the requirement to test initiating a direct connect when the accepting device has an invalid certificate.

Changes:

Checklist Changes

None

Test Plan Changes

[Modify all usages of test 12.5.3.3.2.3 from 135.1-2023 to BTL]

9.9 Data Link Layer - Secure Connect

9.9.6 Is Able to Initiate Direct Connections

The IUT supports initiating direct connections.

…		
~~135.1-2023~~*BTL* - 12.5.3.3.2.3 - Rejection of Invalid Certificate Outgoing Connection Test		
	Test Conditionality	Must be executed.
	Test Directives	~~Repeat with an expired certificate.~~ ~~Repeat with a certificate not signed by the locally configured CA.~~
	Testing Hints	

Specified Test Changes

[Move test 12.5.3.3.2.3 from 135.1-2023 into BTL Specified Tests and modify as shown.]

12.5.3.3.2.3 Rejection of Invalid Certificate Outgoing Connection Test

Reason for Change: Remove requirement to run the test if the accepting device does not have a valid certificate.

Purpose: To verify that the IUT will drop initiated connection attempts if the peer's certificate *has expired* ~~is invalid~~.

Test Concept: With the IUT configured to initiate direct connections. Make the IUT attempt to connect to D3 via a direct connection. D3 presents an *expired* ~~invalid~~ certificate during the *direct connect* connection *attempt*. Verify that the WebSocket is not established.

Configuration Requirements: The IUT is configured to initiate direct connections *to D3*. D3 is configured with *an expiring certificate* ~~an invalid certificate~~. D3 shall be configured to accept the IUT's certificate.

Test Steps:

1. IF (IUT requires D3 to be connected to the hub) THEN
2. MAKE(D3 establish a connection to the hub)
3. WAIT (for D3's certificate to expire)
4. MAKE(the IUT attempt to establish a direct connection to D3)
5. CHECK(that the IUT initiated a WebSocket connection)
6. CHECK(that the WebSocket connection was failed by the IUT)

BTL-26.0 fix3-17: CHANGE_OF_RELIABILITY for EE if Internal Fault Not Supported [BTLWG-1713, CR-0579]

Overview:

Test Plan 5.2.37, Test 8.5.17.9 does not allow the test to be skipped if the EE does not support internal fault. (CR-0579)

Changes:

Checklist Changes

None

Test Plan Changes

5.2 Alarm and Event Management - Notification - Internal - B

[Modify 5.2.37]

5.2.37 Supports CHANGE_OF_RELIABILITY in the Event Enrollment Object

The IUT contains, or can be made to contain, an Event Enrollment object that can generate EventNotifications with an Event_Type of CHANGE_OF_RELIABILITY.

135.1-2023 - 8.5.17.7.1 - Internal Faults Take Precedence Over Monitored Object Faults		
	Test Conditionality	If the IUT does not support an Event Enrollment object which can detect internal faults and monitor an object which detects faults, then this test shall be skipped.
	Test Directives	
	Testing Hints	
135.1-2023 - 8.5.17.7.2 - Monitored Object Faults Take Precedence Over Fault Algorithms		
	Test Conditionality	If the IUT does not support an Event Enrollment object which monitors an object which detects faults and which applies a fault algorithm, then this test shall be skipped.
	Test Directives	
	Testing Hints	
135.1-2023 - 8.5.17.7.3 - Internal Faults Take Precedence Over Fault Algorithms		
	Test Conditionality	If the IUT does not support an Event Enrollment object which can detect internal faults and which applies a fault algorithm, then this test shall be skipped.
	Test Directives	
	Testing Hints	
135.1-2023 - 8.5.17.8 - CHANGE_OF_RELIABILITY of Event Enrollment Object, Monitored Object Fault (UnconfirmedEventNotifications)		
	Test Conditionality	If the IUT has no Event Enrollment object where the *monitored object* ~~Monitored Object that~~ can transition to fault, this test shall be skipped.
	Test Directives	
	Testing Hints	
135.1-2023 - 8.5.17.9 - CHANGE_OF_RELIABILITY of Event Enrollment Object Fault (UnconfirmedEventNotifications)		
	Test Conditionality	~~Must be executed.~~ *If the IUT does not support an Event Enrollment object which supports internal faults, this test shall be skipped.*
	Test Directives	
	Testing Hints	

Specified Test Changes

None

BTL-26.0 fix3-18: Different Source and Destination UDP Ports [BTLWG-1733, CR-0589]

Overview:

Devices must be able to accept, process and correctly respond to a B/IP request that contains a source UDP port different from the destination UDP port.

Changes:

Checklist Changes

None

Test Plan Changes

9.3.2 Is Able to Operate in Normal Mode

…		
135.1-2023 - 12.3.1.9 - Original-Unicast-NPDU		
	Test Conditionality	Must be executed.
	Test Directives	*Repeat this test twice, once with the source UDP port equal to the destination UDP port and once with a different source UDP port.*
	Testing Hints	
…		

9.3.3 Is Able to Operate in Foreign Mode

…		
135.1-2023 - 12.3.1.9 - Original-Unicast-NPDU		
	Test Conditionality	Must be executed.
	Test Directives	*Repeat this test twice, once with the source UDP port equal to the destination UDP port and once with a different source UDP port.*
	Testing Hints	
…		

9.3.4 Is Able to Operate in BBMD Mode

…		
135.1-2023 - 12.3.2.3 - Execute Original-Unicast-NPDU		
	Test Conditionality	Must be executed.
	Test Directives	*Repeat this test twice, once with the source UDP port equal to the destination UDP port and once with a different source UDP port.*
	Testing Hints	
…		

Specified Test Changes

None

BTL-26.0 fix3-19: Clarify When NULL Must Be Allowed [BTLWG-1746, CR-0581]

Overview:

An IR made it clear that only writable non-commandable Present_Value properties are required to accept but not process a write with a NULL value.

Changes:

Checklist Changes

None

Test Plan Changes

4.6Data Sharing - WriteProperty - B

4.6.1 Base Requirements

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

…		
BTL - 9.22.1.X3 - Writing NULL to Non-commandable Properties		
	Test Conditionality	If the IUT claims Protocol_Revision 20, or prior, this test shall be skipped. If the IUT does not contain any writable non-commandable *Present_Value* properties, this test shall be skipped.
	Test Directives	~~Repeat the test for a selection of writable non-commandable, non-Present_Value properties which do not support the value NULL.~~ Repeat the test for each object type with a writable non-commandable Present_Value supported by the IUT which does not support the value NULL.
	Testing Hints	

Specified Test Changes

9.22.1.X3 Writing NULL to Non-commandable Properties

Reason for Change: The standard was changed in PR21 to require that devices not return errors when a RelinquishNULL is written to writable non-commandable Present_Value properties and no test exists for this functionality.

Purpose: This test case verifies that the IUT returns a Result(+) when an attempt is made to relinquish a writable non-commandable Present_Value property.

Test Concept: Write NULL, at a priority, to a writable non-commandable Present_Value property, P1 in object O1, and verify the IUT returns a Result(+) and does not modify the property.

Test Configuration: None.~~P1 shall be a property for which NULL is not an accepted value.~~

Test Steps:

1. READ X = (O1), P1
2. TRANSMIT WriteProperty-Request,
   'Object Identifier' = O1,
   'Property Identifier' = P1,
   'Property Value' = NULL
   'Priority' = (any valid value)

3. RECEIVE BACnet-SimpleACK-PDU
4. VERIFY (O1), P1 = X

BTL-26.0 fix3-20: Fix Test RESTART_AUTONEGOTIATION Command Failure [BTLWG-1741, CR-0587]

Overview:

Based on an IR and CR-0587, the Link_Speed_Autonegotiate property is not required to support the RESTART_AUTONEGOTIATION command.
According to CR-0587, If the device's Network Port does not expose the Link_Speed_Auto-negotiate property, the Test Step 1 should skipped.

Changes:

Checklist Changes

None

Test Plan Changes

[Chage references to Test 7.3.2.46.3.6.2 RESTART_AUTONEGOTIATION Command Failure Test from 135.1-2023 to BTL in the following sections]

9.1.14 Supports the Network Port Object Command Property
9.2.7 Supports the Network Port Object Command Property
9.3.15 Supports the Network Port Object Command Property
9.4.8 Supports the Network Port Object Command Property
9.5.8 Supports the Network Port Object Command Property
9.6.8 Supports the Network Port Object Command Property
9.7.8 Supports the Network Port Object Command Property
9.8.12 Supports the Network Port Object Command Property
9.9.19 Supports the Network Port Object Command Property
9.12.6 Supports the Network Port Object Command Property

Specified Test Changes

[Copy test 7.3.2.46.3.6.2 from 135.1-2023 to BTL and change as shown]

7.3.2.46.3.6.2 RESTART_AUTONEGOTIATION Command Failure Test
Reason for Change: Updated to make the Link_Speed_Autonegotiate check optional

Purpose: To verify that Network Port objects respond to the *RESTART_AUTONEGOTIATION* ~~RESTART_AUTO-NEGOTIATION~~ command with the correct error codes when the command is not supported / enabled.

Test Concept: Starting with a Network Port object which is not configured to auto-negotiate its link speed or which does not support the *RESTART_AUTONEGOTIATION* ~~RESTART_AUTO-NEGOTIATION~~, command it to restart auto-negotiation. Verify that the correct error code is returned.

Configuration Requirements: If the network port support auto-negotiation, disable it. If the IUT does not support the Command property, or all Network Port object support auto-negotiation and it cannot be disabled, then this test shall be skipped.

Test Steps:
-- make sure our initial conditions are good
1. *IF Link_Speed_Autonegotiate is present THEN*
~~VERIFY Link_Speed_Auto-negotiate = TRUE~~
VERIFY Link_Speed_Autonegotiate = TRUE

-- request the renewal, and wait for it to timeout

2. TRANSMIT WriteProperty-Request,
   'Object Identifier' = (the Network Port object),
   'Property Identifier' = Command,
   ~~'Property Value' = RESTART_AUTO_NEGOTIATION~~
   'Property Value' = RESTART_AUTONEGOTIATION

3. IF the port does not support auto-negotiation THEN
   RECEIVE BACnet-Error-PDU
   'Error Class' = PROPERTY,
   'Error Code' = OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED
   ELSE
   RECEIVE BACnet-Error-PDU
   'Error Class' = PROPERTY,
   'Error Code' = VALUE_OUT_OF_RANGE

BTL-26.0 fix3-21: Fix Test 9.2.2.1 to Allow Execution on Every IUT [BTLWG-1742, CR-0586]

Overview:

In its current form, the test cannot be performed with every IUT. (CR586)

Changes:

Checklist Changes

None

Test Plan Changes

[Change all references for test 9.2.2.1 from 135.1-2023 to BTL]

Specified Test Changes

9.2.2.1 Change of Value Notification Arrives after Subscription has Expired
Reason for Change:

Purpose: To verify that an appropriate error is returned if a COV notification arrives after the subscription time period has expired.

Configuration Requirements: If the IUT does not support initiation of SubscribeCOV-Request with 'Issue Confirmed Notifications' equal to TRUE, then this test shall be skipped.

Test Steps:
1. RECEIVE SubscribeCOV-Request,
   'Subscriber Process Identifier' = (any valid process identifier, P1),
   'Monitored Object Identifier' = (any object X of a type that supports COV notification),
   'Issue Confirmed Notifications ' = TRUE,
   'Lifetime' = (any valid Lifetime)
2. TRANSMIT BACnet-SimpleACK-PDU
3. TRANSMIT ConfirmedCOVNotification-Request,
   'Subscriber Process Identifier' = (P1),
   'Initiating Device Identifier' = TD,
   'Monitored Object Identifier' = X,
   'Time Remaining' = (any amount of time greater than 0),
   'List of Values' = (a list of values appropriate to object X)
4. IF (the IUT can cancel the subscription) THEN
   MAKE (the IUT cancel the subscription)
   RECEIVE SubscribeCOV – Request,
   'Subscriber Process Identifier' = (PI),
   'Monitored Object Identifier' = X
   ELSE
   MAKE (the IUT stop resubscribing, if it resubscribes automatically)
5. WAIT (at least Lifetime, but sufficient to ensure the subscription has expired)
6. TRANSMIT ConfirmedCOVNotification-Request,
   'Subscriber Process Identifier' = (P1),
   'Initiating Device Identifier' = TD,
   'Monitored Object Identifier' = X,
   'Time Remaining' = (any amount of time greater than 0),
   'List of Values' = (a list of values appropriate to object X)
7. IF (Protocol_Revision is present and Protocol_Revision >= 10) THEN

```
        RECEIVE BACnet-Error-PDU,
            'Error Class' = SERVICES,
            'Error Code' = UNKNOWN_SUBSCRIPTION |
        (BACnet-SimpleACK-PDU)
    ELSE
        RECEIVE BACnet-Error-PDU,
            'Error Class' = SERVICES,
            'Error Code' = (any valid error code for class SERVICES) |
        (BACnet-SimpleACK-PDU)
```

BTL-26.0 fix3-22: Refine Remaining-Time for Test 12.3.6.3.1 [BTLWG-1671]

Overview:

Making the value for Remaing-Time clearer. Currently the term “test execution time” is used which could be understood for the time since the test started with Step 1. But this is not the value needed for verifying the Remaing-Time as it is requested right now.

Changes:

Checklist Changes

None

Test Plan Changes

[Change all references for test 12.3.6.3.1 from 135.1-2025 to BTL]

Specified Test Changes

12.3.6.3.1 Non-Zero-Duration Foreign Device Table Timer Operations

Reason for change: refining the value for Remaining-Time.

Purpose: To verify that the IUT will handle FDT timer operations: finite time Foreign Device registration, re-registration, adding grace period to the supplied Time-To-Live parameter and FDT entry clearing upon timer expiration.

Configuration Requirements: The TD shall take the role of foreign device FD2. The IUT's FDT must be empty. The Network Port object for the BACnet/IP network is NP.

Notes to Tester: The accuracy of the FDT timer shall be specified by the vendor.

Test Steps:

1. TRANSMIT
   DA = IUT,
   SA = FD2,
   Register-Foreign-Device,
   'Time-To-Live' = 60
2. RECEIVE
   DA = FD2,
   SA = IUT,
   BVLC-Result,
   'Result Code' = 0
3. WAIT (10 seconds)
4. TRANSMIT
   DA = IUT,
   SA = FD2,
   Read-Foreign-Device-Table
5. RECEIVE
   DA = FD2,
   SA = IUT,
   Read-Foreign-Device-Table-Ack,
      B/IP address of FD2,
      Time-To-Live = 60,

Remaining-Time = ~~80 minus test execution time.~~ *90 (Time-To-Live + grace period) - time since FD registration*
-- (50 is also acceptable if Protocol_Revision < 7)

6. IF Protocol_Revision >= 17 THEN
   VERIFY NP, BBMD_Foreign_Device_Table = ((B/IP address of FD2, 60, ~~80~~ *90* - ~~execution time~~ *time since FD registration*))
7. TRANSMIT
   DA = IUT,
   SA = FD2,
   Register-Foreign-Device,
   'Time-To-Live' = 40
8. RECEIVE
   DA = FD2,
   SA = IUT,
   BVLC-Result,
   'Result Code' = 0
9. WAIT (30 seconds)
10. TRANSMIT
    DA = IUT,
    SA = FD2,
    Read-Foreign-Device-Table
11. RECEIVE
    DA = FD2,
    SA = IUT,
    Read-Foreign-Device-Table-Ack,
    B/IP address of FD2, Time-To-Live = 40, Remaining-Time = ~~40 minus test execution time~~ *70 (Time-To-Live + grace period) - time since FD registration*
    -- (10 is also acceptable if Protocol_Revision < 7)
12. IF Protocol_Revision >= 17 THEN
    VERIFY NP, BBMD_Foreign_Device_Table = ((B/IP address of FD2, 40, ~~40~~ *70* - ~~execution time~~ *time since FD registration*))
13. WAIT (50 seconds)
14. TRANSMIT
    DA = IUT,
    SA = FD2,
    Read-Foreign-Device-Table
15. RECEIVE
    DA = FD2,
    SA = IUT,
    Read-Foreign-Device-Table-Ack,
    (No FDT entries)
16. IF Protocol_Revision >= 17 THEN
    VERIFY NP, BBMD_Foreign_Device_Table = ( )

BTL-26.0 fix3-23: Update Test Conditionality for Foreign Mode Testing [BTLWG-1752, CR-0590]

Overview:

The Test Conditionality for 12.3.8.5 too strict and the test should be skipped if the device does not initiate a broadcast on startup when in Foreign Mode?

Changes:

Checklist Changes

None

Test Plan Changes

9.3 Data Link Layer - IPv4

9.3.3 Is Able to Operate in Foreign Mode

The IUT can register as a foreign device with a BBMD.
The IUT supports a configurable BBMD Address to which it sends Register-Foreign-Device NPDU.

…		
135.1-2023 - 12.3.8.5 - Transmits a Broadcast at Startup preceded by Register-Foreign-Device		
	Test Conditionality	If the IUT never transmits a broadcast at startup *while in Foreign Mode*, this test shall be skipped.
	Test Directives	
	Testing Hints	
…		

Specified Test Changes

None