



**BACnet[®] TESTING LABORATORIES
ADDENDA**

**Addendum cov to
BTL Test Package 18.1**

**Revision v3
Revised 11/10/2021**

Approved by the BTL Working Group on October 14, 2021.
Approved by the BTL Working Group Voting Members on November 8, 2021.
Published on November 12, 2021.

[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-18.1-cov-1: Testing Small COV Changes [BTLWG-930].....	2
BTL-18.1-cov-2: UCOV Test Should Validate ProcessID of 0 [BTLWG-953].....	6
BTL-18.1-cov-3: Finite Lifetime Subscriptions [BTLWG-1020]	7
BTL-18.1-cov-4: Datatype DateTime to DS-COVP-B [BTLWG-1027].....	9
BTL-18.1-cov-5: Datatype DateTime to DS-COVM-A [BTLWG-1036]	10
BTL-18.1-cov-6: Datatype DateTime to DS-COVM-B [BTLWG-1037]	11
BTL-18.1-cov-7: Changes to Test Confirmed Change of Value Notification from Property Value [BTLWG-1039]	12
BTL-18.1-cov-8: Clarification for test Canceling COV Subscriptions [BTLWG-1094].....	14
BTL-18.1-cov-9: COV on Status_Flags Conditionality [BTLWG-956, CR-0012].....	15

In the following document, language to be added to existing clauses within the BTL Test Package 18.1 is indicated through the use of *italics*, while deletions are indicated by ~~striketrough~~. 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-18.1-cov-1: Testing Small COV Changes [BTLWG-930, CR-0219]

Overview:

CR-0219 raised the Question: Should the test be modified to allow for the testing of objects where any change will result in a COV notification? The issue is that: In objects that have a fixed COV_Increment that is equal to the smallest change that the object supports, these steps cannot be executed as written.

Any COV tests in DS-COV-B, DS-COVP-B, and DS-COVM-B which expect to change the monitored value by less than COV_Increment need to be modified to take CR-0219 into account. A similar treatment in all of these tests is here proposed.

Changes:

BTL Specified Test Changes

[Modify clause 8.2.1 of BTL Specified Tests]

8.2 ConfirmedCOVNotification Service Initiation Tests

8.2.1 Change of Value Notification from ~~an Analog Input, Analog Output, and Analog Value~~ a Numeric Object's Present_Value Property

Reason for Change: Add more primitive value objects. Updated description of the 'List of Values' to improve readability. Updated 'Configuration Requirements'. Add clarification to test that the last COVNotification shall reflect the correct values.

Purpose: To verify that the IUT can initiate ConfirmedCOVNotification service requests conveying a change of the Present_Value property of Analog Input, Analog Output, *Lighting Output*, ~~and~~ Analog Value, *Large Analog Value*, *Integer Value*, and *Positive Integer Value* 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 from the set Analog Input, Analog Output, *Lighting Output*, ~~and~~ Analog Value, *Large Analog Value*, *Integer Value*, and *Positive Integer Value*) 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

```

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 initial Present_Value and initial Status_Flags)
4. TRANSMIT BACnet-SimpleACK-PDU
5. TRANSMIT ReadProperty-Request,
  'Object Identifier' = X,
  'Property Identifier' = COV_Increment
6. RECEIVE BACnet-ComplexACK-PDU,
  'Object Identifier' = X,
  'Property Identifier' = COV_Increment,
  'Property Value' = (a value "increment" that will be used below)
7. IF (Out_Of_Service is writable) THEN
  WRITE X, Out_Of_Service = TRUE
RECEIVE BACnet-SimpleACK-PDU
  BEFORE Notification Fail Time
    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 initial ReportedPV = the current Present_Value, and new
Status_Flags)
    TRANSMIT BACnet-SimpleACK-PDU
8. IF (Present_Value is now writable) THEN
  WRITE X, Present_Value = (any value that differs from "initial Present_Value" ReportedPV by less than
"increment")
RECEIVE BACnet-SimpleACK-PDU
  ELSE
    MAKE (Present_Value = any value that differs from "initial Present_Value" ReportedPV by less than "increment")
9. WAIT Notification Fail Time
10. CHECK (verify that no COV notification was transmitted)
11. IF (Present_Value is now writable) THEN
  WRITE X, Present_Value = (any value that differs from "initial Present_Value" ReportedPV by an amount greater
than "increment")
RECEIVE BACnet-SimpleACK-PDU
  ELSE
    MAKE (Present_Value = any value that differs from "initial Present_Value" ReportedPV by an amount greater than
"increment")
12. BEFORE NotificationFailTime
  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)
13. TRANSMIT BACnet-SimpleACK-PDU
14. TRANSMIT SubscribeCOV-Request,
  'Subscriber Process Identifier' = (the same value used in step 1),
  'Monitored Object Identifier' = X
15. RECEIVE BACnet-SimpleACK-PDU
16. IF (Out_Of_Service is writable) THEN
  WRITE X, Out_Of_Service = FALSE
RECEIVE BACnet-SimpleACK-PDU

```

[Modify clause 9.11.1.9 of BTL Specified Tests]

9.11 SubscribeCOVProperty Service Execution Tests

9.11.1 Positive SubscribeCOVProperty Service Execution Tests

9.11.1.9 Client-Supplied COV Increment

Reason for Change: Modify the test to work with all numeric datatypes.

Purpose: To verify that the IUT correctly generates COV notifications when the client supplies the COV increment in the SubscribeCOVProperty request. Either confirmed or unconfirmed notifications may be used but at least one of these options must be supported by the IUT.

Test Concept: A subscription for COV notification is made for a property of *numeric* datatype ~~REAL~~. The subscription request specifies a COV increment. The monitored property is changed by an amount less than the increment, and the *tester* ~~TD~~ waits to ensure that the IUT does not generate a notification. The monitored property is changed by an amount slightly more than is required to cause a COV notification, and the *tester observes* ~~TD waits for~~ the notification.

Test Configuration Requirements: If the property being subscribed to has a related COV_Increment property in the object, then the value of the COV_Increment property should be significantly different than the COV increment provided in the subscription service. *In devices where the 'COV_Increment' is always less than the minimal change that the monitored property can make, skip steps 4 and 5.*

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
 - 'Subscriber Process Identifier' = (any valid process identifier),
 - 'Monitored Object Identifier' = (any object supporting COV notifications),
 - 'Issue Confirmed Notifications' = TRUE | FALSE,
 - 'Lifetime' = (any value that will ensure no re-subscription is required to complete the test),
 - 'Monitored Property Identifier' = (any valid property supporting COV notifications),
 - 'COV Increment' = (any valid increment value)
2. RECEIVE BACnet-SimpleACK-PDU
3. BEFORE **Notification Fail Time**
 - IF (the subscription was for confirmed notifications) THEN
 - RECEIVE BACnetConfirmedCOVNotification-Request,
 - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
 - 'Initiating Device Identifier' = IUT,
 - 'Monitored Object Identifier' = (the same object used in the subscription),
 - 'Time Remaining' = (the requested lifetime),
 - 'List of Values' = (values appropriate to the object type of the monitored object including the value of monitored property)
 - TRANSMIT BACnet-SimpleACK-PDU
 - ELSE
 - RECEIVE BACnetUnconfirmedCOVNotification-Request,
 - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
 - 'Initiating Device Identifier' = IUT,
 - 'Monitored Object Identifier' = (the same object used in the subscription),
 - 'Time Remaining' = (the requested lifetime),
 - 'List of Values' = (values appropriate to the object type of the monitored object including the value of monitored property)
4. MAKE (the monitored property change by less than the COV increment)
5. CHECK (verify that the IUT did not transmit a notification message for the monitored property)
6. MAKE (the monitored property change by slightly more than COV Increment less the amount changed in step 54)
7. BEFORE **Notification Fail Time**
 - IF (the subscription was for confirmed notifications) THEN
 - RECEIVE BACnetConfirmedCOVNotification-Request,
 - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
 - 'Initiating Device Identifier' = IUT,
 - 'Monitored Object Identifier' = (the same object used in the subscription),
 - 'Time Remaining' = ?,
 - 'List of Values' = (values appropriate to the object type of the monitored object

```
        including the changed value that triggered the notification)
    TRANSMIT BACnet-SimpleACK-PDU
ELSE
    RECEIVE BACnetUnconfirmedCOVNotification-Request,
        'Subscriber Process Identifier' = (the same identifier used in the subscription),
        'Initiating Device Identifier' = IUT,
        'Monitored Object Identifier' = (the same object used in the subscription),
        'Time Remaining' = ?,
        'List of Values' = (values appropriate to the object type of the monitored object
                           including the changed value that triggered the notification)
```

BTL-18.1-cov-2: UCOV Test Should Validate ProcessID of 0 [BTLWG-953, CR-0077]**Overview:**

CR-0077 included several typographical errors in several tests and one functional correction in Test 8.3.9. Due to the age of the clarification request most of the issues have been fixed. However, 8.3.9 went through some correction, a similar test was also added to BTL Specified Tests (8.3.X11).

Changes:

BTL Test Plan Changes

[In BTL Testplan, modify the following testplan entry in section 4.18.1 Base Requirements]

BTL - 8.3.X11 - Unsubscribed COV Service Initiation Test 8.3.9 Unsubscribed Change of Value Notifications		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

BTL Specified Test Changes

[Remove 8.3.X11 - Unsubscribed COV Service Initiation Test from BTL specified Tests]

[Move 8.3.9 Unsubscribed Change of Value Notifications into BTL Specified Tests and modify]

8.3.9 Unsubscribed Change of Value Notifications

Reason for Change: Add Process ID Requirement and Abort Conditionality to test.

Unsubscribed COV notifications differ from subscribed COV notifications that use the UnconfirmedCOVNotification service in two respects. First, no subscription is required. Second, the 'Subscriber Process Identifier' parameter usually has a value of zero.

BACnet Reference Clauses: 13.7.

Purpose: To verify that the IUT can initiate UnconfirmedCOVNotification service requests when no subscription for the COV notification has been made.

Test Concept: The IUT is configured to send unsubscribed COV notifications. The TD then waits for the notification. Given that there is no defined trigger, the vendor shall inform the tester how to make the IUT generate the notifications if they are not sent periodically.

Test Steps:

1. MAKE (the IUT send an unsubscribed COV notification)

2. BEFORE Notification Fail Time

RECEIVE UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = 0,
 'Initiating Device Identifier' = IUT,
 'Monitored Object Identifier' = (any valid object identifier),
 'Time Remaining' = 0,
 'List of Values' = (any valid properties and values from the monitored object)

BTL-18.1-cov-3: Finite Lifetime Subscriptions [BTLWG-1020, CR-0480]**Overview:**

Correcting the typo error: If (the subscription was for confirmed notifications) then before Notification Fail Time IUT will receive BACnetConfirmedCOVNotification-Request instead BACnetUnconfirmedCOVNotification-Request, however test steps 9 had typo error. Per CR-0480.

Changes:

BTL Specified Test Changes

[In BTL Specified Tests 16.1, 9.11.1.7 Finite Lifetime Subscriptions, Page No. 279]

9.11.1.7 Finite Lifetime Subscriptions

Reason for change: Updates description of 'Time Remaining' and adds validation that this value counts down as expected.

Correcting the typo error at step 9 and updated the language as per BACnet 135.1 6 clause

Purpose: To verify that the IUT correctly responds to a SubscribeCOVProperty request to establish a subscription with a temporary lifetime. Either confirmed or unconfirmed notifications may be used, but at least one of these options must be supported by the IUT.

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
 - 'Subscriber Process Identifier' = *PIDI*, any valid process identifier),
 - 'Monitored Object Identifier' = *OI*, any object supporting COV notifications),
 - 'Issue Confirmed Notifications' = TRUE | FALSE,
 - 'Lifetime' = *TI*, a *any* value between 60 seconds and 300 seconds),
 - 'Monitored Property Identifier' = *PI*, any valid property supporting COV notifications)
2. RECEIVE BACnet-SimpleACK-PDU
3. ~~BEFORE Notification Fail Time~~
3. IF (the subscription was for confirmed notifications) THEN
 - BEFORE Notification Fail Time*
 - RECEIVE BACnetConfirmedCOVNotification-Request,
 - 'Subscriber Process Identifier' = *PIDI*(the same identifier used in the subscription),
 - 'Initiating Device Identifier' = IUT,
 - 'Monitored Object Identifier' = *OI*(the same object used in the subscription),
 - 'Time Remaining' = (the requested subscription lifetime *A value approximately equal to, but not greater than TI*),
 - 'List of Values' = (values appropriate to the property subscribed to, and any other properties the IUT provides with it, such as Status-Flags)
 - TRANSMIT BACnet-SimpleACK-PDU
- ELSE
 - BEFORE Notification Fail Time*
 - RECEIVE BACnetUnconfirmedCOVNotification-Request,
 - 'Subscriber Process Identifier' = *PIDI*(the same identifier used in the subscription),
 - 'Initiating Device Identifier' = IUT,
 - 'Monitored Object Identifier' = *OI*(the same object used in the subscription),
 - 'Time Remaining' = ~~(the requested lifetime)~~ = *(A value approximately equal to, but not greater than TI)*,
 - 'List of Values' = (values appropriate to the property subscribed to, and any other properties the IUT provides with it, such as Status-Flags)
4. MAKE (a change to the monitored object that should causes a COV notification)
5. *WAIT a period longer than the resolution of the IUT's COV subscription lifetime timer*
5. ~~BEFORE Notification Fail Time~~
6. IF (the subscription was for confirmed notifications) THEN
 - BEFORE Notification Fail Time*
 - RECEIVE BACnetConfirmedCOVNotification-Request,
 - 'Subscriber Process Identifier' = *PIDI*(the same identifier used in the subscription),
 - 'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = *OI*(the same object used in the subscription),
 'Time Remaining' = (*T2*: a value greater than 0 and less than the requested subscription lifetime),
 'List of Values' = (values appropriate to the property subscribed to, and any other
 properties the IUT provides with it, such as Status-Flags)
 TRANSMIT BACnet-SimpleACK-PDU
 ELSE
 BEFORE Notification Fail Time
 RECEIVE BACnetUnconfirmedCOVNotification-Request,
 'Subscriber Process Identifier' = *PID1*(the same identifier used in the subscription),
 'Initiating Device Identifier' = IUT,
 'Monitored Object Identifier' = *OI*(the same object used in the subscription),
 'Time Remaining' = (*T2*: a value greater than 0 and less than the requested subscription lifetime),
 'List of Values' = (values appropriate to the object type of the monitored object
 including the changed value that triggered the notification)
 6. ~~WAIT (the lifetime of the subscription)~~
 7. WAIT a period longer than the resolution of the IUT's COV subscription lifetime timer
 8. MAKE (a change to the monitored object that causes a COV notification)
 9. IF (the subscription was for confirmed notifications) THEN
 BEFORE Notification Fail Time
 RECEIVE *BACnetConfirmedCOVNotification-Request*,
 'Subscriber Process Identifier' = *PID1*,
 'Initiating Device Identifier' = IUT,
 'Monitored Object Identifier' = *OI*,
 'Time Remaining' = (a value greater than 0 and less than the *TRT2*),
 'List of Values' = (values appropriate to the object type of the monitored object)
 ELSE
 BEFORE Notification Fail Time
 RECEIVE BACnetUnconfirmedCOVNotification-Request,
 'Subscriber Process Identifier' = *PID1*,
 'Initiating Device Identifier' = IUT,
 'Monitored Object Identifier' = *OI*,
 'Time Remaining' = (a value greater than 0 and less than the *TRT2*),
 'List of Values' = (values appropriate to the object type of the monitored object
 including the changed value that triggered the notification)
 10. WAIT (the lifetime of the subscription)
 11. MAKE (a change to the monitored object that would cause a COV notification if there were an active subscription)
 12. CHECK (verify that the IUT did not transmit a COV notification message)

BTL-18.1-cov-4: Datatype DateTime to DS-COVP-B [BTLWG-1027]**Overview:**

In the current checklist and Test Plan, datatype DateTime in DS-COVP-B is missing.

Changes:

BTL Checklist Changes

[In BTL Checklist, add entry for DateTime at the end of current list]

Data Sharing - Change Of Value Property - B		

	C^2	<i>Supports COVP for DateTime properties</i>
¹ At least one of these options is required in order to claim conformance to this BIBB.		
² At least one of these options is required in order to claim conformance to this BIBB.		

BTL Test Plan Changes

[In BTL Test Plan, Add new section to 4.20 Data Sharing - Change Of Value Property - B]

4.20.X Supports COVP for DateTime Properties

The IUT supports change of value notifications for at least one DateTime property value.

BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value		
	Test Conditionality	Must be executed.
	Test Directives	Apply to at least 1 property of the specified datatype. Ensure that after all applications of this test (regardless of the property datatype it is applied for), that the test has been applied at least once to each object type which supports COVP on one or more of its properties
	Testing Hints	
BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value		
	Test Conditionality	Must be executed.
	Test Directives	Apply to at least 1 property of the specified datatype. Ensure that after all applications of this test (regardless of the property datatype it is applied for), that the test has been applied at least once to each object type which supports COVP on one or more of its properties
	Testing Hints	

BTL-18.1-cov-5: Datatype DateTime to DS-COVM-A [BTLWG-1036]**Overview:**

In the current checklist and Test Plan, datatype DateTime in DS-COVM-A is missing.

Changes:

BTL Checklist Changes

[In BTL Checklist, add entry for DateTime at the end of current list]

Data Sharing - Change Of Value Multiple - A		

	C^3	<i>Can subscribe to DateTime property values</i>
¹ At least one of these options is required in order to claim conformance to this BIBB. ² At least one of these options is required in order to claim conformance to this BIBB. ³ At least one of these options is required in order to claim conformance to this BIBB.		

BTL Test Plan Changes

[In BTL Test Plan, Add new section to 4.25 Data Sharing - Change Of Value Multiple - A]

4.25.X Can Subscribe to DateTime Property Values

The IUT can subscribe for, receive and process Change of Value Multiple notifications from DateTime property values.

BTL - 8.X12.1.7 - Change of Value Multiple Notifications		
	Test Conditionality	Must be executed.
	Test Directives	Execute test using 'Monitored Object' = (any valid object containing a property to monitor and which the vendor supports in a SubscribeCOVPropertyMultiple-Request) and 'Monitored Property Identifier' = (any property which the vendor supports in a SubscribeCOVPropertyMultiple-Request that can contain a DateTime value)
	Testing Hints	

BTL-18.1-cov-6: Datatype DateTime to DS-COVM-B [BTLWG-1037]**Overview:**

In the current checklist and Test Plan, datatype DateTime in DS-COVM-B is missing.

Changes:

BTL Checklist Changes

[In BTL Checklist, add entry for DateTime at the end of current list]

Data Sharing - Change Of Value Multiple - B		

	C^2	<i>Supports COVM for DateTime property values</i>
¹ At least one of these options is required in order to claim conformance to this BIBB.		
² At least one of these options is required in order to claim conformance to this BIBB.		

BTL Test Plan Changes

[In BTL Test Plan, Add new section to 4.26 Data Sharing - Change Of Value Multiple - B]

4.26.X Can Subscribe to DateTime Property Values

The IUT supports change of value notifications for at least one DateTime property values

BTL - 9.X41.1.3 - Confirmed Change of Value Notification From Property Value		
	Test Conditionality	Must be executed.
	Test Directives	Apply to at least 1 property of the specified datatype. Ensure that after all applications of this test (regardless of the property datatype it is applied for), that the test has been applied at least once to each object type which supports COVM on one or more of its properties.
	Testing Hints	
BTL - 9.X41.1.4 - Unconfirmed Change of Value Notification From Property Value		
	Test Conditionality	Must be executed.
	Test Directives	Apply to at least 1 property of the specified datatype. Ensure that after all applications of this test (regardless of the property datatype it is applied for), that the test has been applied at least once to each object type which supports COVM on one or more of its properties.
	Testing Hints	

BTL-18.1-cov-7: Changes to Test Confirmed Change of Value Notification from Property Value [BTLWG-1039]

Overview:

In the BTL Specified Tests-18.1_final.pdf, test case 9.11.1.X11 Confirmed Change of Value Notification from Property Value : step 6 Simple ACK is missing
This proposal will include this missing test step

Changes:

BTL Specified Test Changes

[In BTL Specified Test, 9.11.1.X11 make the following change]

9.11.1.X11 Confirmed Change of Value Notification from Property Value

Reason for Change: No test exists for this functionality. This test is not in any SSPC proposal.

Purpose: To verify that the IUT can initiate ConfirmedCOVNotification service requests conveying a change of the Property Value.

Test Concept: A property 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 Value of the monitored Property is changed, and a notification shall be received. The subscribed property may be changed using the WriteProperty service or by another means. 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.

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
 'Subscriber Process Identifier' = (any valid process identifier),
 'Monitored Object Identifier' = X
 'Issue Confirmed Notifications' = TRUE,
 'Lifetime' = L
 'Monitored Property Identifier' = Y (any valid property supporting COV notifications)
2. RECEIVE BACnet-SimpleACK-PDU
3. BEFORE Notification Fail Time
 RECEIVE BACnetConfirmedCOVNotification-Request,
 'Subscriber Process Identifier' = (the same identifier used in the subscription),
 'Initiating Device Identifier' = IUT,
 'Monitored Object Identifier' = X
 'Time Remaining' = (any value appropriate for the Lifetime selected),
 'List of Values' = (values appropriate to the property subscribed to, and any other properties the IUT provides with it, such as Status-Flags)
4. TRANSMIT BACnet-SimpleACK-PDU
5. MAKE (a change to the monitored object PROPERTY that causes a COV notification)
6. BEFORE Notification Fail Time
 RECEIVE BACnetConfirmedCOVNotification-Request,
 'Subscriber Process Identifier' = (the same identifier used in the subscription),
 'Initiating Device Identifier' = IUT,
 'Monitored Object Identifier' = X
 'Time Remaining' = (any value appropriate for the Lifetime selected),
 'List of Values' = (values appropriate to the property subscribed to, and any other properties the IUT provides with it, such as Status-Flags)
7. TRANSMIT BACnet-SimpleACK-PDU
87. TRANSMIT SubscribeCOVProperty-Request,
 'Subscriber Process Identifier' = (the same identifier used in Step 1),
 'Monitored Object Identifier' = X
 'Monitored Property Identifier' = Y

98. RECEIVE BACnet-SimpleACK-PDU

BTL-18.1-cov-8: Clarification for test Canceling COV Subscriptions [BTLWG-1094]

Overview:

Currently 9.11.1.4 requires that 9.11.1.1 be run before it but does not make this clear. It also has requirements in the Note to Tester which should be in the test steps.

Changes:

BTL Specified Tests Changes

[In BTL Specified Tests, move test 9.11.1.4 from 135.1-2019 into the BTL Specified Tests, and modify]

9.11.1.4 Canceling COV Subscriptions

Reason for Change: Added Configuration Requirements and check at end of test to remove the Notes to Tester requirement.

Dependencies: Indefinite lifetime COV subscriptions, 9.11.1.1.

Purpose: To verify that the IUT correctly responds to a SubscribeCOVProperty request to cancel a COV subscription. This test cancels the subscription made in 9.11.1.1.

Configuration Requirements: This test should be executed after test 9.11.1.1, while the subscription created in that test still exists in the IUT.

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
 - 'Subscriber Process Identifier' = (the process identifier used in test 9.11.1.1),
 - 'Monitored Object Identifier' = (the same object used in test 9.11.1.1),
 - 'Monitored Property Identifier' = (the same property used in test 9.11.1.1)
2. RECEIVE BACnet-SimpleACK-PDU
3. WAIT Notification Fail Time
4. MAKE (a change to the monitored object that would cause a COV notification if there were an active subscription)
5. *CHECK(the IUT does not transmit a COV notification)*

~~Notes to Tester: The IUT shall not transmit a COV notification message.~~

BTL-18.1-cov-9: COV on Status_Flags Conditionality [BTLWG-956, CR-0012]**Overview:**

CR-0012 indicated that the Conditionality for COV tests for changes in Status_Flags should be conditional on the Status_Flags being changeable without changing Present_Value to achieve it.

Changes:**BTL Checklist Changes**

[In BTL Checklist, modify Test Conditionality of Status_Flags changes test and renumber footnotes]

Data Sharing - Change Of Value Property - B		
	R	Base Requirements
	R	Supports Lifetimes up to 8 hours in duration
	R C ¹	Supports COVP for Status_Flags changes
	C ² € ¹	Supports COVP for non-array properties
	C ² € ¹	Supports COVP for non-array properties
	C ² € ¹	Supports COVP for array elements
	C ² € ¹	Supports COVP for the size of an array
	C ² € ¹	Supports COVP for whole arrays
	O	Supports COVP for list property
	C ³ € ²	Supports COVP for NULL property values
	C ³ € ²	Supports COVP for BOOLEAN property values
	C ³ € ²	Supports COVP for Enumerated property values
	C ³ € ²	Supports COVP for INTEGER property values
	C ³ € ²	Supports COVP for Unsigned property values
	C ³ € ²	Supports COVP for REAL property values
	C ³ € ²	Supports COVP for Double property values
	C ³ € ²	Supports COVP for Time property values
	C ³ € ²	Supports COVP for Date property values
	C ³ € ²	Supports COVP for CharacterString property values
	C ³ € ²	Supports COVP for OctetString property values
	C ³ € ²	Supports COVP for BitString property values
	C ³ € ²	Supports COVP for BACnetObjectIdentifier property values
	C ³ € ²	Supports COVP for constructed property values
	C ³ € ²	Supports COVP for Value Source properties
¹ Required if the Status_Flags property ever changes value in an object which supports COV property subscriptions. ²¹ At least one of these options is required in order to claim conformance to this BIBB. ³² At least one of these options is required in order to claim conformance to this BIBB.		

BTL Test Plan Changes

[Change every occurrence of 8.2.8 to reference 8.2.2]

[Change every occurrence of 8.3.8 to reference 8.3.2]

[Change every occurrence of 8.2.2 in the Test Plan]

BTL - 8.2.2 - Change of Value Notification for Changes to Status_Flags Property		
	Test Conditionality	This may be skipped if 8.3.2 is executed against an <object type> object. <i>This test shall be skipped if the Status_Flags property cannot be changed or can only be changed as a side-effect of changing Present_Value.</i>
	Test Directives	The selected object must be an <object type>.
	Testing Hints	

[Change every occurrence of 8.3.2 in the Test Plan]

BTL - 8.3.2 - Change of Value Notification from a Numeric Object's Status_Flags Property		
	Test Conditionality	This may be skipped if 8.2.2 is executed against an <object type> object. <i>This test shall be skipped if the Status_Flags property cannot be changed or can only be changed as a side-effect of changing Present Value.</i>
	Test Directives	The selected object must be an <object type>.
	Testing Hints	

[Change Test Plan section 4.20.3]

4.20.3 Supports COVP for Status_Flags Changes

The IUT supports change of value notifications for Status_Flags changes.

BTL - 9.11.1.X21 - Confirmed Change of Value Notification from Status_Flags Property		
	Test Conditionality	<i>This test shall be skipped if no objects support:</i> <ul style="list-style-type: none"> - the Status_Flags property, - changes in Status_Flags other than via changes in Present_Value and, - SubscribeCOVProperty. Must be executed if object type contains a Status_Flag and property which supports SubscribeCOVProperty.
	Test Directives	Repeat test for at least one object of each type <i>that meets the test conditionality.</i>
	Testing Hints	
BTL - 9.11.1.X22 - Unconfirmed Change of Value Notification from Status_Flags Property		
	Test Conditionality	<i>This test shall be skipped if no objects support:</i> <ul style="list-style-type: none"> - the Status_Flags property, - changes in Status_Flags other than via changes in Present_Value and, - SubscribeCOVProperty. Must be executed if object type contains a Status_Flag and property which supports SubscribeCOVProperty.
	Test Directives	Repeat test for at least one object of each type <i>that meets the test conditionality.</i>
	Testing Hints	

BTL Specified Test Changes

[Add deletion of 8.2.8 and 8.3.8 into BTL specified tests]

The tests 8.2.8 and 8.3.8, with the changes above to the Test Plan, are now obsolete and not used. This section notes that these are put into BTL Specified Tests and labeled for deletion in the 135.1.