



**BACnet<sup>®</sup> TESTING LABORATORIES  
ADDENDA**

**Addendum cr1 to  
BTL Test Package 23.3**

**Revision final  
Revised 3/25/2024**

Approved by the BTL Working Group on February 8, 2024.  
Approved by the BTL Working Group Voting Members on March 20, 2024.  
Published on March 29, 2024.

**[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-23.3 cr1-1: Acked Transitions Test Fixes [BTLWG-1106. CR-0496].....	2
BTL-23.3 cr1-2: Allow Unicast WhoHas [BTLWG-1254. CR-0524] .....	4
BTL-23.3 cr1-3: Improve Out_of_Service Test Language [BTLWG-1289. CR-0519] .....	7
BTL-23.3 cr1-4: Add Forwarded-NPDU in Foreign Mode [BTLWG-1291. CR-0529].....	9
BTL-23.3 cr1-5: Remove DISABLE parameter from Tests [BTLWG-1350. CR-0537] .....	10
BTL-23.3 cr1-6: Event_Detection_Enable Testing Update [BTLWG-1392. CR-0417] .....	11
BTL-23.3 cr1-7: Add Time Duration Parameter [BTLWG-1403. CR-0544] .....	13
BTL-23.3 cr1-8: Fix Result Flags for Test 9.21.1.6 [BTLWG-1419. CR-0550] .....	15
BTL-23.3 cr1-9: Improve the Testing of WPM [BTLWG-1436. CR-0545] .....	16
BTL-23.3 cr1-10: Simplify Test Purpose for Commandable Objects [BTLWG-1440. CR-0515].....	19
BTL-23.3 cr1-11: Fix Elapsed Active Time Test [BTLWG-1443] .....	20
BTL-23.3 cr1-12: Fix 10.2.X Tests to Ensure Correct SINFO Noted [BTLWG-1452. CR-0555].....	22
BTL-23.3 cr1-13: Loosen Lifetime Restrictions on Subscribe COV Test [BTLWG-1460. CR-0557] .....	28

In the following document, language to be added to existing clauses within the BTL Test Package 23.3 is indicated through the use of *italics*, while deletions are indicated by ~~strike through~~. 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-23.3 cr1-1: Acked Transitions Test Fixes [BTLWG-1106. CR-0496]**

**Overview:**

Test 7.3.1.11.1 does not account for objects which only do normal to normal transitions (CR-0496)

**Changes:**

---

**Checklist Changes**

---

None

---

**Test Plan Changes**

---

[Add new test to section 5.5: Alarm and Event management - Acknowledge - B]

...	
<b>BTL - 7.3.1.11.X1 - Acked Transitions Test for Normal-To-Normal Transitions</b>	
<b>Test Conditionality</b>	Must be executed if the IUT supports normal to normal event generating objects.
<b>Test Directives</b>	
<b>Testing Hints</b>	

---

**Specified Test Changes**

---

[Add test 7.3.1.11.X1]

**7.3.1.11.X1 Acked\_Transitions Test for Normal-To-Normal Transitions**

Reason for Change: No test exists for this functionality.

Purpose: To verify that the Acked\_Transitions property tracks whether an acknowledgment has been received for a previously issued normal event notification. It also verifies the interrelationship between Status\_Flags and Event\_State.

Test Concept: The IUT is configured such that the Event\_Enable property indicates that normal event transitions are to trigger an event notification. The normal event transition is triggered and the Acked\_Transitions property is monitored to verify that the appropriate bit is cleared when a notification message is transmitted and reset when an acknowledgment is received.

Configuration Requirements: The Event\_Enable and Acked\_Transitions properties shall be configured with a value of (?, ?, TRUE). The referenced event-triggering property shall be set to a value that results in a NORMAL condition. The value of the Transitions parameter for all recipients shall be (?, ?, TRUE).

Notes to Tester: The UnconfirmedEventNotification service may be substituted for the ConfirmedEventNotification service, in which case the TD shall skip sending the BACnet-SimpleACK-PDU messages after receiving the notification.

Notes to Tester: For life safety objects that latch pMonitoredValue, the LifeSafetyOperation Service will be required to reset pMonitoredValue.

**Test Steps:**

1. VERIFY pCurrentState = NORMAL
2. VERIFY Acked\_Transitions = (?, ?, TRUE)
3. IF (Protocol\_Revision is present AND Protocol\_Revision >= 13) THEN  
     VERIFY Status\_Flags = (FALSE, FALSE, ?, ?)
4. IF (pMonitoredValue is writable) THEN

```

WRITE pMonitoredValue = (a value that will result in a TO_NORMAL transition)
ELSE
    MAKE (pMonitoredValue a value that will result in a TO_NORMAL transition)
5. WAIT (pTimeDelayNormal)
6. BEFORE Notification Fail Time
    RECEIVE ConfirmedEventNotification-Request,
        'Process Identifier' = (PI2: any valid process ID),
        'Initiating Device Identifier' = IUT,
        'Event Object Identifier' = (the event-generating object configured for this test),
        'Time Stamp' = (Tnormal: any valid time stamp),
        'Notification Class' = (the class corresponding to the object being tested),
        'Priority' = (Pnormal: the value configured to correspond to a TO-NORMAL transition),
        'Event Type' = (any valid event type),
        'Message Text' = (optional, any valid message text),
        'Notify Type' = (the notify type configured for this event),
        'AckRequired' = TRUE,
        'From State' = NORMAL,
        'To State' = NORMAL,
        'Event Values' = (values appropriate to the event type)
7. TRANSMIT BACnet-SimpleACK-PDU
8. VERIFY pCurrentState = NORMAL
9. VERIFY Acked_Transitions = (?, ?, FALSE)
10. IF (Protocol_Revision is present AND Protocol_Revision >= 13) THEN
    VERIFY Status_Flags = (FALSE, FALSE, ?,?)
11. TRANSMIT AcknowledgeAlarm-Request,
    'Acknowledging Process Identifier' = (PI2),
    'Event Object Identifier' = (the event-generating object configured for this test),
    'Event State Acknowledged' = NORMAL,
    'Time Stamp' = (Tnormal),
    'Acknowledgement Source' = (a character string),
    'Time of Acknowledgment' = (the TD's current time)
12. RECEIVE BACnet-SimpleACK-PDU
13. IF (Protocol_Revision is present AND Protocol_Revision >= 1) THEN
    BEFORE Notification Fail Time
        RECEIVE ConfirmedEventNotification-Request,
            'Process Identifier' = (PI2),
            'Initiating Device Identifier' = IUT,
            'Event Object Identifier' = (the event-generating object configured for this test),
            'Time Stamp' = (the IUT's current time or sequence number),
            'Notification Class' = (the class corresponding to the object being tested),
            'Priority' = (Pnormal),
            'Event Type' = (any valid event type),
            'Message Text' = (optional, any valid message text),
            'Notify Type' = ACK_NOTIFICATION,
            'To State' = NORMAL
    ELSE
        BEFORE Notification Fail Time
            RECEIVE ConfirmedEventNotification-Request,
                'Process Identifier' = (PI2),
                'Initiating Device Identifier' = IUT,
                'Event Object Identifier' = (the event-generating object configured for this test),
                'Time Stamp' = (the IUT's current time or sequence number),
                'Notification Class' = (the class corresponding to the object being tested),
                'Priority' = (Pnormal),
                'Event Type' = (any valid event type),
                'Message Text' = (optional, any valid message text),
                'Notify Type' = ACK_NOTIFICATION
14. TRANSMIT BACnet-SimpleACK-PDU
15. VERIFY Acked_Transitions = (?, ?, TRUE)

```

## BTL-23.3 cr1-2: Allow Unicast WhoHas [BTLWG-1254. CR-0524]

### Overview:

The BACnet standard (per addendum 135-2012ar-5) now allows the IUT to send and receive a unicast response. Per CR-0524 - Change tests 8.32.1 - 8.32.4 to allow unicast WhoHas messages.

### Changes:

---

## Checklist Changes

---

None

---

## Test Plan Changes

---

[Change references to 8.32.1, 8.32.2, 8.32.3, 8.32.4 from 135.1-2023 to BTL]

---

## Specified Test Changes

---

[Move tests 8.32.1, 8.32.2, 8.32.3, 8.32.4 from 135.1-202d into BTL Specified Tests and modified as shown below.]

### 8.32.1 Object Identifier Selection with no Device Instance Range

Reason for Change: The BACnet standard (per addendum 135-2012ar-5) now allows the IUT to send and receive a unicast response.

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

Notes to Tester: If there is no vendor-defined observable action, then test step 3 can be skipped.

Test Steps:

1. RECEIVE  
DESTINATION = **TD** | LOCAL BROADCAST | GLOBAL BROADCAST  
SOURCE = IUT,  
Who-Has-Request,  
'Object Identifier' = Object1
2. TRANSMIT  
DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST  
SOURCE = TD,  
I-Have-Request,  
'Device Identifier' = (the TD's Device object)  
'Object Identifier' = Object1
3. CHECK (for any vendor-defined observable actions)

### 8.32.2 Object Name Selection with no Device Instance Range

Reason for Change: The BACnet standard (per addendum 135-2012ar-5) now allows the IUT to send and receive a unicast response.

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

Notes to Tester: If there is no vendor-defined observable action, then test step 3 can be skipped.

Test Steps:

1. RECEIVE  
DESTINATION = **TD** | LOCAL BROADCAST | GLOBAL BROADCAST,  
SOURCE = IUT,

- Who-Has-Request,  
'Object Name' = V1
- 2. TRANSMIT  
DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST  
SOURCE = TD,  
I-Have-Request,  
'Device Identifier' = (the TD's Device object)  
'Object Name' = V1
- 3. CHECK (for any vendor-defined observable actions)

### 8.32.3 Object Identifier Selection with a Device Instance Range

Reason for Change: The allowance for Unicast I-Have is added.

Purpose: To verify that the IUT can initiate Who-Has service requests using the object identifier form with a device instance range. If the IUT cannot be caused to issue a Who-Has request of this form, then this test shall be omitted.

Notes to Tester: Device instance range should be selected to cover TD's device object identifier. If there is no vendor-defined observable action, then test step 3 can be skipped.

Test Steps:

- 1. RECEIVE  
DESTINATION = **TD** | LOCAL BROADCAST | GLOBAL BROADCAST,  
SOURCE = IUT,  
Who-Has-Request,  
'Device Instance Range Low Limit' = (any integer X:  $40 \leq X \leq$  'Device Instance Range High Limit'),  
'Device Instance Range High Limit' = (any integer Y: 'Device Instance Range Low Limit'  $\leq Y \leq$  4,194,303),  
'Object Identifier' = Object1
- 2. TRANSMIT  
DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST  
SOURCE = TD,  
I-Have-Request,  
'Device Identifier' = (the TD's Device object)  
'Object Identifier' = Object1
- 3. CHECK (for any vendor-defined observable actions)

### 8.32.4 Object Name Selection with a Device Instance Range

Reason for Change: The allowance for Unicast I-Have is added.

Purpose: To verify that the IUT can initiate Who-Has service requests using the object name form with a device instance range. If the IUT cannot be caused to issue a Who-Has request of this form, then this test shall be omitted.

Notes to Tester: Device instance range should be selected to cover TD's device object identifier. If there is no vendor-defined observable action, then test step 3 can be skipped.

Test Steps:

- 1. RECEIVE  
DESTINATION = **TD** | LOCAL BROADCAST | GLOBAL BROADCAST,  
SOURCE = IUT,  
Who-Has-Request,  
'Device Instance Range Low Limit' = (any integer X:  $40 \leq X \leq$  'Device Instance Range High Limit'),  
'Device Instance Range High Limit' = (any integer Y: 'Device Instance Range Low Limit'  $\leq Y \leq$  4,194,303),  
'Object Name' = V1
- 2. TRANSMIT  
DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST  
SOURCE = TD,  
I-Have-Request,  
'Device Identifier' = (the TD's Device object)  
'Object Name' = V1
- 3. CHECK (for any vendor-defined observable actions)



## BTL-23.3 cr1-3: Improve Out\_of\_Service Test Language [BTLWG-1289. CR-0519]

### Overview:

Improve the language for the Out\_Of\_Service for Commandable Value Objects Test.

### Changes:

---

## Checklist Changes

---

None

---

## Test Plan Changes

---

[Modify references to test 7.3.1.1.2 from 135.1-2023 to BTL]

---

## Specified Test Changes

---

[Move test 7.3.1.1.2 from 135.1-2023 into BTL Specified Tests and modify has shown below

### 7.3.1.1.2 Out\_Of\_Service for Commandable Value Objects Test

Reason for change: To improve the language of steps 1 and 5 to make it clear that in step 1 the change in Present\_Value succeeds and that in step 5, the attempted change in Present\_Value would not be expected to succeed.

Purpose: To verify that Present\_Value is no longer updated by software local to the IUT when Out\_Of\_Service is TRUE.

**Test Concept:** Select an object whose Present\_Value is being modified by software local to the IUT at Priority PTY1. The value of the Out\_Of\_Service property is set to TRUE, the Present\_Value property is written at PTY1 and the Present\_value is checked to ensure the Present\_Value is no longer being modified by software local to the IUT.

**Test Concept:** An object's Present\_Value, at a priority, PTY1, is being controlled by a process, P1, internal to the IUT. P1 triggers a change to Present\_Value when Out\_Of\_Service is FALSE and Present\_Value is verified to change. Out\_Of\_Service is then set to TRUE and Present\_Value read. P1 executes such that a change to Present\_Value would occur and Present\_Value is verified to be unchanged.

**Configuration Requirements:** The values of the entries in the Priority\_Array above PTY1 shall be NULL. *The test starts with Out\_Of\_Service = FALSE.*

**Notes to Tester:** The specifics of the MAKE steps are defined by the vendor. They may require tester interaction or simply wait while P1 executes.

### Test Steps:

1. MAKE P1 (Present\_Value = PV1, any valid value, using software local to the IUT change the Present\_Value at PTY1)
2. CHECK (Present\_Value changed)
3. IF (Out\_Of\_Service is writable) THEN  
    WRITE Out\_Of\_Service = TRUE  
    ELSE  
        MAKE (Out\_Of\_Service TRUE)
34. VERIFY READ PVI = Present\_Value
4. WRITE Present\_Value, PTY1 = PV2, any valid value other than PV1
5. VERIFY Present\_Value = PV2
5. MAKE P1 (Present\_Value = PV3, any valid value other than PV2, using software local to the IUT attempt to change the Present\_Value at PTY1 to a value other than PVI)
6. VERIFY Present\_Value = PV2 PVI



**BTL-23.3 cr1-4: Add Forwarded-NPDU in Foreign Mode [BTLWG-1291. CR-0529]**

**Overview:**

Add new test for Forwarded NPDU while in Foreign Device Mode.

**Changes:**

---

**Checklist Changes**

---

None

---

**Test Plan Changes**

---

[In Test plan, change section 9.3.3 - Change reference from 12.3.1.10 to use new test 12.3.8.X2]

...	
<b>BTL - 12.3.8.X2 - Forwarded-NPDU (Two-hop Distribution) in Foreign Mode</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	
...	

---

**Specified Test Changes**

---

[In BTL Specified Tests, add test 12.3.8.X2]

**12.3.8.X2 Forwarded-NPDU (Two-hop Distribution) in Foreign Mode**

Reason For Change: The specific case of processing a Forwarded-NPDU message while in foreign mode did not exist.

Purpose: To verify that an IUT, configured in foreign mode, will process a Forwarded-NPDU message from a BBMD.

Configuration Requirements: The IUT is registered as a Foreign Device with the TD. The TD is configured as a BBMD, on a different IP subnet than the IUT.

**Test Steps:**

1. TRANSMIT DA = SA = TD,  
Forwarded-NPDU,  
Originating-Device = D1,  
NPDU = Who-Is
2. IF (the IUT responds with Unicast I-Am) THEN  
RECEIVE DA = D1,  
Original-Unicast-NPDU,  
NPDU = I-Am  
ELSE  
RECEIVE DA = TD, SOURCE = IUT,  
Distribute-Broadcast-To-Network-NPDU,  
NPDU = I-Am
3. CHECK (The IUT shall not issue any Forwarded-NPDUs)

## BTL-23.3 cr1-5: Remove DISABLE parameter from Tests [BTLWG-1350. CR-0537]

### Overview:

Modify the tests for ReinitializeDevice that use the deprecated parameter DISABLE.

### Changes:

---

## Checklist Changes

---

None

---

## Test Plan Changes

---

[Change references to tests 8.24.2, 8.24.3, and 8.24.6 from 135.1-2023 to BTL]

---

## Specified Test Changes

---

[Move these tests from 135.1 into BTL Specified Tests and modify as noted.]

### 8.24.2 Indefinite Duration, Disable, Password

Reason For Change: The test is requesting the deprecated value DISABLE for the 'Enable/Disable' parameter.

Purpose: To verify that the IUT can initiate DeviceCommunicationControl service requests that indicate communication should cease for an indefinite time duration and convey a password.

Test Steps:

1. RECEIVE DeviceCommunicationControl-Request,  
'Enable/Disable' = ~~DISABLE~~ **DISABLE-INITIATION**,  
'Password' = (a password of up to 20 characters)
2. TRANSMIT BACnet-SimpleACK-PDU

### 8.24.3 Time Duration, Disable, Password

Reason For Change: The test is requesting the deprecated value DISABLE for the 'Enable/Disable' parameter.

Purpose: To verify that the IUT can initiate DeviceCommunicationControl service requests that indicate communication should cease for a specific time duration and convey a password.

Test Steps:

1. RECEIVE DeviceCommunicationControl-Request,  
'Time Duration' = (any unsigned value > 0),  
'Enable/Disable' = ~~DISABLE~~ **DISABLE-INITIATION**,  
'Password' = (a password of up to 20 characters)
2. TRANSMIT BACnet-SimpleACK-PDU

### 8.24.6 Time Duration, Disable, No Password

Reason For Change: The test is requesting the deprecated value DISABLE for the 'Enable/Disable' parameter.

Purpose: To verify that the IUT can initiate DeviceCommunicationControl service requests that indicate communication should cease for a specific time duration and do not convey a password. If the IUT does not support the "no password" option, this test shall not be performed.

Test Steps:

1. RECEIVE DeviceCommunicationControl-Request,  
'Time Duration' = (any unsigned value > 0),  
'Enable/Disable' = ~~DISABLE~~ **DISABLE-INITIATION**,
2. TRANSMIT BACnet-SimpleACK-PDU

**BTL-23.3 cr1-6: Event\_Detection\_Enable Testing Update [BTLWG-1392. CR-0417]**

**Overview:**

CR-0417 clearly allows for Event\_Detection\_Enable to be read-only and TRUE. The subsequent work item (BTLWG-320) did not cover make this clarification clear to the tester.

The Test Conditionality needs to allow for the test to be skipped if Event\_Detection\_Enable cannot be made FALSE.

**Changes:**

---

**Checklist Changes**

---

None

---

**Test Plan Changes**

---

[Modify section 5.2.1 Base Requirements for AE-N-I-B, Modify section 5.22.1 Base Requirements for AE-LS-B and 5.32.1 Base Requirements for AE-AC-B]

<b>BTL - 7.3.1.22.1 - Event_Detection_Enable Inhibits Event Generation</b>	
<b>Test Conditionality</b>	If Protocol Revision < 13, then this test shall be skipped. If Event_Detection_Enable cannot be set to FALSE, this test shall be skipped.
<b>Test Directives</b>	The objects selected by the tester should include all variants that differ in the set of supported alarming properties, or the writability of any of those properties. At least one instance of each variant should be selected.
<b>Testing Hints</b>	

---

**Specified Test Changes**

---

[In BTL Specified Tests, modify test 7.3.1.22.1 with the following changes]

**7.3.1.22.1 Event\_Detection\_Enable Inhibits Event Generation**

Reason for Change: Remove expectation that Event\_Detection\_Enable is able to be equal to both TRUE and FALSE as per CR-0417.

Purpose: To verify that Event\_Detection\_Enable enables and disables event detection in objects which are configured for event reporting

Test Concept: Select an event generating object, O1, that is configured for supports event reporting. If possible, make the object generate an event, to an offnormal, so that if the object can have a non-normal state, it enters that state early in the test. This will help detect incorrect implementations that initiate a TO\_NORMAL transition when the algorithm is disabled. Set the Event\_Detection\_Enable property to FALSE. Verify the Event\_State is NORMAL and the Acked\_Transitions, Event\_Time\_Stamps, and Event\_Message\_Texts are equal to their respective initial conditions, as mandated in the standard. Repeat the process that made the object generate an event and observe that no notification messages are transmitted. If possible, Make a condition exist that would cause a transition if event reporting were enabled and observe that no notification messages are transmitted.

Configuration Requirements: O1 is configured with Event\_Detection\_Enable set to FALSE. If Event\_Detection\_Enable cannot be set to FALSE, this test shall be skipped. DELAY shall represent the time delay appropriate to the transition being tested (i.e. Time\_Delay for TO\_OFFNORMAL, 0 for TO\_FAULT and FAULT to NORMAL transitions, and either Time\_Delay or Time\_Delay Normal for TO\_NORMAL) to detect and report unconfirmed events and requires acknowledgments for all transitions. Event\_Detection\_Enable is equal to TRUE. D1 is either the pTimeDelay, or pTimeDelayNormal parameter, or 0 (for transitions to and from FAULT state) depending on the event transition. For this

test, NO\_TS equals a BACnetDateTime with all unspecified values, a BACnet Time with all unspecified values, or a sequence number of 0.

Test Steps:

1. ~~VERIFY Event\_Detection\_Enable = FALSE~~
1. ~~VERIFY Event\_Detection\_Enable = TRUE~~
2. ~~MAKE (a condition exist which will cause O1 to transition, to an offnormal state if possible)~~
3. ~~WAIT D1~~
4. ~~**BEFORE Notification Fail Time**~~
  - ~~RECEIVE UnconfirmedEventNotificationRequest~~
  - ~~'Process Identifier' = \_\_\_\_\_ (any valid process identifier);~~
  - ~~'Initiating Device Identifier' = IUT;~~
  - ~~'Event Object Identifier' = O1;~~
  - ~~'Time Stamp' = \_\_\_\_\_ (any valid time stamp);~~
  - ~~'Notification Class' = \_\_\_\_\_ (the notification class configured for O1);~~
  - ~~'Priority' = \_\_\_\_\_ (the value configured for the transition);~~
  - ~~'Event Type' = \_\_\_\_\_ (any valid event type);~~
  - ~~'Message Text' = \_\_\_\_\_ (optional, any valid message text);~~
  - ~~'Notify Type' = \_\_\_\_\_ (value from the Notify\_Type property configured for O1);~~
  - ~~'AckRequired' = TRUE;~~
  - ~~'From State' = \_\_\_\_\_ (any valid event state);~~
  - ~~'To State' = \_\_\_\_\_ (any event state appropriate to the event type);~~
  - ~~'Event Values' = \_\_\_\_\_ (any values appropriate to the event type)~~
5. ~~IF Event\_Detection\_Enable is writable THEN~~
  - ~~WRITE Event\_Detection\_Enable = FALSE~~
  - ~~ELSE~~
    - ~~MAKE (Event\_Detection\_Enable to FALSE. This property is expected to be set during system configuration and is not expected to change dynamically.)~~
6. ~~WAIT (D1 + **Notification Fail Time** + **Internal Processing Fail Time**)~~
7. ~~CHECK (that the IUT did not send any further event notifications for O1)~~
82. ~~VERIFY pCurrentState = NORMAL~~
93. ~~VERIFY Acked\_Transitions = (T,T,T)~~
104. ~~IF (Protocol\_Revision is present AND Protocol\_Revision = 1) THEN~~
  - ~~VERIFY Event\_Time\_Stamps = [NO\_TS, NO\_TS, NO\_TS ]~~
115. ~~IF (Event\_Message\_Texts property exists) THEN~~
  - ~~VERIFY Event\_Message\_Texts = ["", ""]~~
126. ~~MAKE (a condition exist which would cause O1 to transition, if Event\_Detection\_Enable were TRUE)~~
137. ~~WAIT (D1 + **DELAY** + **Notification Fail Time**)~~
148. ~~CHECK (that the IUT did not send any event notifications for O1)~~
159. ~~VERIFY pCurrentState = NORMAL~~
1610. ~~VERIFY Acked\_Transitions = (T,T,T)~~
1711. ~~IF (Protocol\_Revision is present AND Protocol\_Revision = 1) THEN~~
  - ~~VERIFY Event\_Time\_Stamps = [NO\_TS, NO\_TS, NO\_TS]~~
1812. ~~IF (Event\_Message\_Texts property exists) THEN~~
  - ~~VERIFY Event\_Message\_Texts = ["", ""]~~

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

## BTL-23.3 cr1-7: Add Time Duration Parameter [BTLWG-1403. CR-0544]

### Overview:

Add the time duration parameter to DeviceCommunicationControl testing. Devices must accept a limited time duration and can optionally support indefinite time duration.

### Changes:

---

## Checklist Changes

---

None

---

## Test Plan Changes

---

[Change references to tests 9.24.2.1 and 9.24.2.2 from 135.1-2023 to BTL]

---

## Specified Test Changes

---

[Move these tests from 135.1 into BTL Specified Tests and modify as noted.]

### 9.24.2.1 Invalid Password

Reason for Change: Added Time Duration to step 1 as finite time duration is required and infinite time duration is optional. Removed step 3 as not required for this test.

Purpose: To verify the correct execution of DeviceCommunicationControl service procedure when an invalid password is provided. ~~If the IUT does not provide password protection this test case shall be omitted.~~

*Test Concept: With the IUT and TD communicating, transmit a DeviceCommunicationControl service using DISABLE\_INITIATION and an invalid password. Verify the IUT responds with the correct error class and code.*

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request,  
*'Time Duration' = (a value  $T \geq 1$ , in minutes)*  
'Enable/Disable' = DISABLE\_INITIATION,  
'Password' = (any invalid password)
2. RECEIVE BACnet-Error-PDU,  
Error Class = SECURITY,  
Error Code = PASSWORD\_FAILURE
- ~~3. VERIFY (Device, X), System\_Status = (any valid value)~~

### 9.24.2.2 Missing Password

Reason for Change: Added Time Duration to step 1 as finite time duration is required and infinite time duration is optional. Removed step 3 as not required for this test.

Purpose: To verify the correct execution of DeviceCommunicationControl service procedure when a password is required but not provided. ~~If the IUT does not provide password protection, then this test case shall be omitted.~~

*Test Concept: With the IUT and TD communicating, transmit a DeviceCommunicationControl service using DISABLE\_INITIATION and no password. Verify the IUT responds with one of the valid error classes and codes.*

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request,  
*'Time Duration' = (a value  $T \geq 1$ , in minutes)*  
'Enable/Disable' = DISABLE\_INITIATION

2. IF (Protocol\_Revision >= 7) THEN  
    RECEIVE BACnet-Error-PDU,  
    Error Class = SECURITY,  
    Error Code = PASSWORD\_FAILURE  
ELSE  
    (RECEIVE BACnet-Error-PDU,  
    Error Class = SECURITY,  
    Error Code = PASSWORD\_FAILURE)  
    | (RECEIVE BACnet-Error-PDU,  
    Error Class = SERVICES,  
    Error Code = MISSING\_REQUIRED\_PARAMETER)  
3. ~~VERIFY (Device, X), System\_Status = (any valid value)~~

## BTL-23.3 cr1-8: Fix Result Flags for Test 9.21.1.6 [BTLWG-1419. CR-0550]

### Overview:

BTL-CR-0550 point out that the 'Results Flags' parameter specified in Step 2 of the test 9.21.1.6 is incorrect.

### Changes:

---

## Checklist Changes

---

None

---

## Test Plan Changes

---

[Change references to test 9.21.1.6 from 135.1-2023 to BTL]

---

## Specified Test Changes

---

[Move these tests from 135.1 into BTL Specified Tests and modify as noted.]

### 9.21.1.6 Reading a Range of Items that do not Exist *by Position*

Reason for Change: Fix the Result Flags parameter in step 2.

Purpose: To verify that the IUT correctly responds to a ReadRange service request when there are no items within the specified by position range.

Test Concept: A ReadRange request is transmitted by the TD requesting a range of items all known not to be in the list property P. The IUT shall respond by returning an empty list.

Configuration Requirements: The list property, P, is configured with N items.

Test Steps:

1. TRANSMIT ReadRange-Request,
  - 'Object Identifier' = (the object configured for this test),
  - 'Property Identifier' = P,
  - 'Reference Index' = (any value x: x > N),
  - 'Count' = (any value y: y > 0)
2. RECEIVE Read-Range-ACK,
  - 'Object Identifier' = (the object configured for this test),
  - 'Property Identifier' = P,
  - 'Result flags' = {TRUE, FALSE, TRUE, FALSE, FALSE},
  - 'Item Count' = 0,
  - 'Item Data' = (an empty list)

**BTL-23.3 cr1-9: Improve the Testing of WPM [BTLWG-1436. CR-0545]**

**Overview:**

Testplan Clause 4.7.26 tests that a IUT correctly writes constructed values that the IUT supports. This clause indicates all forms of constructed values should be tested including all forms of ABSTRACT-SYNTAX.&Type.

Note, if an IUT is a B-AWS is must support writing all options and choices. It is not required to write all possible variants of an ABSTRACT-SYNTAX.&Type. The advanced modify BIBBs cover off any additional writes the IUT must support.

Of the properties that contain the ABSTRACT-SYNTAX.&Type type, only a few are writable and of those, only the Action property of the Command object is not limited in the standard.

**Proposed solution for Writable properties containing ABSTRACT-SYNTAX.&Type:**

- Command object -> Action property -> property-value (**test primitive values only**) (Although not explicitly stated in the standard, writes using primitive values would be the common application)
- Timer object -> State\_Change\_Values property -> constructed-value (**no testing required**) (limited to non-primitive values in the standard) (has explicit primitive and lighting-command options)
- All objects -> Tags property -> value (**test primitive values and BACnetDateTime**) (limited to primitive values and BACnetDateTime in the standard)
- Schedule object -> Present\_Value property (**test primitive values only**) (limited to primitive values only in the standard)
- Schedule object -> Weekly\_Schedule property -> day-schedule -> value (**test primitive values only**) (limited to primitive values only in the standard)
- Schedule object -> Exception\_Schedule property -> BACnetSpecialEvent -> list-of-time-values -> value (**test primitive values only**) (limited to primitive values only in the standard)

**Changes:**

---

**Checklist Changes**

---

None

---

**Test Plan Changes**

---

**4.5.21 Can Write Constructed Property Values**

The IUT is able to write constructed values to properties. Constructed values are SEQUENCES or CHOICES of basic or other constructed datatypes.

<b>135.1-2019 - 8.22.1 - Writing Non-Array Properties, or 135.1-2019 - 8.22.2 - Writing Array Properties</b>	
<b>Test Conditionality</b>	<del>For each type of standard constructed datatypes that the IUT can write. At least one of these tests must be applied. At least one of the tests (8.22.1 .. 8.22.2) shall be executed against a property with a constructed value.</del>
<b>Test Directives</b>	<i>This test shall be repeated for each standard constructed value that the IUT is able to write.</i>  <i>1. Where a constructed value can take on different forms, such as a constructed value that contains optional elements, or is a CHOICE, the tester should test all supported forms of the datatype.</i> <i>2. Where the constructed value contains an ANY type, the tester should limit testing to supported primitive datatypes and, if supported, BACnetDateTime.</i>

	<p>3. Where the constructed value contains a <i>SEQUENCE OF</i>, the tester should limit testing to 0, 1, 2, and 3 members and the maximum number of members, if defined.</p> <p>4. Where the constructed value contains a <i>SEQUENCE</i>, the value is tested with several iterations that are limited to:</p> <p style="padding-left: 40px;"><i>Iteration 1: all REQUIRED and OPTIONAL parameters in the SEQUENCE,</i></p> <p style="padding-left: 40px;"><i>Iteration 2: all REQUIRED and a subset of OPTIONAL parameters in the SEQUENCE,</i></p> <p style="padding-left: 40px;"><i>Iteration 3: only REQUIRED parameters in the SEQUENCE,</i></p> <p style="padding-left: 40px;"><i>Iteration 4: if the SEQUENCE contains a CHOICE, follow Hint #1,</i></p> <p style="padding-left: 40px;"><i>Iteration 5: if the SEQUENCE contains an ANY type, follow Hint #2,</i></p> <p style="padding-left: 40px;"><i>Iteration 6: if the SEQUENCE contains a SEQUENCE OF, follow Hint #3.</i></p> <p><i>Note: After a complex datatype or ANY type has been validated as part of one constructed value, testing of other constructed values that contain the same datatype or ANY type can be limited to tests that ensure the IUT correctly generates valid writes.</i></p>
<b>Testing Hints</b>	

### 4.7.26 Can Write Constructed Property Values

The IUT is able to write constructed property values.

<p><b>135.1-2019 - 8.23.1 - Writing a Single Property of a Single Object,</b>  <b>135.1-2019 - 8.23.2 - Writing Multiple Properties of a Single Object,</b>  <b>135.1-2019 - 8.23.3 - Writing Multiple Objects, One Property Each, or</b>  <b>135.1-2019 - 8.23.4 - Writing Multiple Objects, Multiple Properties for Each</b></p>	
<b>Test Conditionality</b>	<p>At least one of the tests (8.23.1 .. 8.23.4) shall be executed against a property with a constructed value. <del>This test shall be repeated for each standard constructed value that the IUT is able to write.</del></p>
<b>Test Directives</b>	<p><del>At least one of the properties written by the selected test shall contain a constructed value. This test shall be repeated for each standard constructed value that the IUT is able to write.</del></p> <ol style="list-style-type: none"> <li>1. Where a constructed value can take on different forms, such as a constructed value that contains optional elements, or is a CHOICE, the tester should test all supported forms of the datatype.</li> <li>2. Where the constructed value contains an ANY type, the tester should limit testing to supported primitive datatypes and, if supported, BACnetDateTime.</li> <li>3. Where the constructed value contains a SEQUENCE OF, the tester should limit testing to 0, 1, 2, and 3 members and the maximum number of members, if defined.</li> <li>4. Where the constructed value contains a SEQUENCE, the value is tested with several iterations that are limited to:             <ul style="list-style-type: none"> <li><i>Iteration 1: all REQUIRED and OPTIONAL parameters in the SEQUENCE,</i></li> <li><i>Iteration 2: all REQUIRED and a subset of OPTIONAL parameters in the SEQUENCE,</i></li> <li><i>Iteration 3: only REQUIRED parameters in the SEQUENCE,</i></li> <li><i>Iteration 4: if the SEQUENCE contains a CHOICE, follow Hint #1,</i></li> <li><i>Iteration 5: if the SEQUENCE contains an ANY type, follow Hint #2,</i></li> </ul> </li> </ol>

	<p><i>Iteration 6: if the SEQUENCE contains a SEQUENCE OF, follow Hint #3.</i></p> <p><i>Note: After a complex datatype or ANY type has been validated as part of one constructed value, testing of other constructed values that contain the same datatype or ANY type can be limited to tests that ensure the IUT correctly generates valid writes.</i></p>
<b>Testing Hints</b>	

---

## Specified Test Changes

---

None

## BTL-23.3 cr1-10: Simplify Test Purpose for Commandable Objects [BTLWG-1440. CR-0515]

### Overview:

The test specification from 135.1-2019 is very specific to certain object types and needs to be generalized to cover all the objects that are commandable.

### Changes:

---

## Checklist Changes

---

None

---

## Test Plan Changes

---

None

---

## Specified Test Changes

---

[In BTL Specified Test, Change test 7.3.1.2 Relinquish Default Test purpose to be more general.]

### 7.3.1.2 Relinquish Default Test

Reason for Change: Add validation of Current\_Command\_Priority property. **Simplify the test purpose to include all commandable objects.**

Purpose: To verify that the Present\_Value property takes on the value of Relinquish\_Default when all prioritized commands have been relinquished. This test applies to **Analog Output, Analog Value, Binary Output, Binary Value, Multi-state Output, and Multi-state Value** objects that are commandable.

Test Concept: A pre-requisite to this test is that an object has been provided for which all prioritized commands have been relinquished and any minimum on/off time has been accounted for. The Present\_Value is compared to the value of Relinquish\_Default to ensure that they are the same. If possible, the value of Relinquish\_Default is changed to verify that Present\_Value tracks the changes.

Configuration Requirements: The object to be tested shall be configured such that all slots in the Priority\_Array have a value of NULL and no internal algorithms are issuing prioritized commands to this object.

#### Test Steps:

1. VERIFY Priority\_Array = (NULL, NULL, NULL)
2. IF Protocol\_Revision >= 17 THEN {  
     VERIFY Current\_Command\_Priority = NULL  
   }
3. READ X = Present\_Value
2. TRANSMIT ReadProperty Request,  
     'Object Identifier' = (the object being tested),  
     'Property Identifier' = Present\_Value
3. RECEIVE ReadProperty ACK,  
     'Object Identifier' = (the object being tested),  
     'Property Identifier' = Present\_Value  
     'Property Value' = (any valid value, X)
4. VERIFY Relinquish\_Default = X
5. IF (Relinquish\_Default is writable) THEN {  
     WRITE Relinquish\_Default = (any valid value, Y, other than X ~~the one returned in step 3~~)  
     VERIFY Present\_Value = Y  
   }

## BTL-23.3 cr1-11: Fix Elapsed Active Time Test [BTLWG-1443]

### Overview:

The math used to calculate the total elapsed time is incorrect and needs to be fixed.

### Changes:

---

## Checklist Changes

---

None

---

## Test Plan Changes

---

[Change references to test 7.3.1.9 from 135.1-2023 to BTL]

---

## Specified Test Changes

---

[Move 7.3.1.9 from 135.1-2023 into BTL Specified Tests and modify line 14 as shown.]

### 7.3.1.9 Elapsed Active Time Test

Reason for the change: Correct calculation error in step 14.

Purpose: To verify that the properties of objects that collectively track active time function properly.

Test Concept: The Present\_Value or Feedback\_Value of the object being tested is set to INACTIVE. The Elapsed\_Active\_Time property is checked to verify that it does not accumulate time while the object is in an INACTIVE state. The Present\_Value or Feedback\_Value is then set to ACTIVE. The Elapsed\_Active\_Time property is checked to verify that it is accumulating time while the object is in an ACTIVE state. The Elapsed\_Active\_Time is reset. The Time\_Of\_Active\_Time\_Reset property is checked to verify that it has been updated.

Configuration Requirements: The object being tested shall be configured such that the Present\_Value or Feedback\_Value if that is used for the calculation, and Elapsed\_Active\_Time properties are writable or another means of changing these properties shall be provided. Whether Present\_Value or Feedback\_Value is used as the indicator for the calculation of the Elapsed\_Active\_Time is a local matter.

Notes To Tester: It was intentional to specify that the alternative use of Feedback\_Value tracking specified in 135-2010ad-3 is allowed regardless of the Protocol\_Revision claimed by the implementation.

### Test Steps:

1. IF (Present\_Value is writable) THEN  
    WRITE Present\_Value = INACTIVE  
    VERIFY Present\_Value = INACTIVE  
ELSE  
    MAKE (Present\_Value = INACTIVE)
2. IF (Feedback\_Value is used for Elapsed\_Active\_Time tracking) THEN  
    WAIT(long enough for Feedback\_Value to reflect the Present\_Value)  
    VERIFY Feedback\_Value = INACTIVE
3. READ Elapsed\_Active\_Time = initialElapsedTime  
  
-- verify that Elapsed\_Active\_Time does not change when the object is INACTIVE
4. WAIT (more than Internal\_Processing Fail Time + at least 1 second)
5. VERIFY Elapsed\_Active\_Time = initialElapsedTime  
  
-- verify that Elapsed\_Active\_Time correctly reflects the time the object is ACTIVE5
6. IF (Present\_Value is writable) THEN

```

WRITE Present_Value = ACTIVE
VERIFY Present_Value = ACTIVE
ELSE
  MAKE (Present_Value = ACTIVE)
7. IF (Feedback_Value is used for Elapsed_Active_Time tracking) THEN
  WAIT (long enough for Feedback_Value to reflect the Present_Value)
  VERIFY Feedback_Value = ACTIVE
8. READ initialTime = (the IUT's Device object) Local_Time
9. WAIT (more than Internal Processing Fail Time + 30 seconds)
10. IF (Present_Value is writable) THEN
  WRITE Present_Value = INACTIVE
  VERIFY Present_Value = INACTIVE
  ELSE
  MAKE (Present_Value = INACTIVE)
11. IF (Feedback_Value is used for Elapsed_Active_Time tracking) THEN
  WAIT (long enough for Feedback_Value to reflect the Present_Value)
  VERIFY Feedback_Value = INACTIVE
12. READ currentTime = (the IUT's Device object) Local_Time
13. READ totalElapsedTime = Elapsed_Active_Time
14. CHECK (totalElapsedTime == (currentTime - initialTime) - initialElapsedTime)
14. CHECK (totalElapsedTime == (currentTime - initialTime) + initialElapsedTime)

-- verify ability to reset Elapsed_Active_Time, if it is writable
15. IF (Elapsed_Active_Time is writable) THEN
  WRITE Elapsed_Active_Time = 0
  READ currentDate = (the IUT's Device object) Local_Date
  READ currentTime = (the IUT's Device object) Local_Time
  VERIFY Time_Of_Active_Time_Reset == { currentDate, currentTime }

```

## BTL-23.3 cr1-12: Fix 10.2.X Tests to Ensure Correct SINFO Noted [BTLWG-1452. CR-0555]

### Overview:

CR-0555 pointed out that at least one test (10.2.2.5.1) had a test step that was missing SNET and SADR information. This work item was created to review other 10.2.x tests to check for missing SNET and SADR components.

### Changes:

---

## Checklist Changes

---

None

---

## Test Plan Changes

---

[Change references to tests 10.2.2.4.4, 10.2.2.5.1, 10.2.2.5.2, 10.2.6, 10.2.2.4.2, 10.2.3.6.2 from 135.1-2023 to BTL]

---

## Specified Test Changes

---

[Move tests 10.2.2.4.4, 10.2.2.5.1, 10.2.2.5.2, 10.2.6, 10.2.2.4.2, 10.2.3.6.2 into BTL Specified Tests, and modify]

### 10.2.2.4.4 Timeout

Reason for Change: Correct tests to specify correct SINFO information.

Purpose: To verify that the IUT restores the availability status of DNETs after the busy timer expires.

### Test Steps:

1. TRANSMIT PORT B,  
DESTINATION = LOCAL BROADCAST,  
SOURCE = R2-3,  
Router-Busy-To-Network,  
Network Numbers = 3
2. RECEIVE PORT A,  
DESTINATION = LOCAL BROADCAST,  
SOURCE = IUT,  
Router-Busy-To-Network,  
Network Numbers = 3
3. WAIT (30 seconds)
4. TRANSMIT PORT A,  
DA = IUT,  
SOURCE = D1A,  
DNET = 3,  
DADR = D3D,  
Hop Count = 255,  
ReadProperty-Request,  
'Object Identifier' = (any BACnet standard object),  
'Property Identifier' = (any required property of the specified object)
5. RECEIVE PORT B,  
DA = R2-3,  
SOURCE = IUT,  
SNET = I,  
SADR = DIA,  
DNET = 3,  
DADR = D3D,  
Hop Count = (any integer x: 0 < x < 255),

ReadProperty-Request,  
'Object Identifier' = (the object identifier used in step 4),  
'Property Identifier' = (the property identifier used in step 4)

#### 10.2.2.5.1 Restoring Specific DNETs

Reason for Change: Correct test to specify correct SINFO information. Add new Notes to Tester.

Purpose: To verify that the IUT updates its network availability information when a Router-Available-To-Network message conveying specific DNETs is received.

*Notes to Tester: This test is to be run after test 10.2.2.2.5 such that R2-3 is known to be a router to network 6.*

Test Steps:

1. TRANSMIT PORT B,  
DESTINATION = LOCAL BROADCAST,  
SOURCE = R2-3,  
Router-Busy-To-Network
2. RECEIVE PORT A,  
DESTINATION = LOCAL BROADCAST,  
SOURCE = IUT,  
Router-Busy-To-Network,  
Network Numbers = 3, 6 | 6, 3
3. TRANSMIT PORT B,  
DESTINATION = LOCAL BROADCAST,  
SOURCE = R2-3,  
Router-Available-To-Network,  
Network Numbers = 3
4. RECEIVE PORT A,  
DESTINATION = LOCAL BROADCAST,  
SOURCE = IUT,  
Router-Available-To-Network,  
Network Numbers = 3
5. TRANSMIT PORT A,  
DESTINATION = IUT,  
SOURCE = D1A,  
DNET = 3,  
DADR = D3D,  
Hop Count = 255,  
ReadProperty-Request,  
'Object Identifier' = (any BACnet standard object),  
'Property Identifier' = (any required property of the specified object)
6. RECEIVE PORT B,  
DESTINATION = R2-3,  
SOURCE = IUT,  
SNET = 1,  
SADR = D1A,  
DNET = 3,  
DADR = D3D,  
Hop Count = (any integer x: 0 < x < 255),  
ReadProperty-Request,  
'Object Identifier' = (the object identifier used in step 5),  
'Property Identifier' = (the property identifier used in step 5)
7. TRANSMIT PORT A,  
DESTINATION = IUT,  
SOURCE = D1A,  
DNET = 6,  
DADR = (any valid device address),  
Hop Count = 255,  
ReadProperty-Request,

'Object Identifier' = (any BACnet standard object),  
'Property Identifier' = (any required property of the specified object)

8. RECEIVE PORT A,  
DESTINATION = D1A,  
SOURCE = IUT,  
Reject-Message-To-Network,  
Reject Reason = 2 (router busy),  
DNET = 6

#### 10.2.2.5.2 Restoring All DNETs

Reason for Change: Correct test to specify correct SINFO information. Add new Notes to Tester.

Purpose: To verify that the IUT updates its network availability information when a Router-Available-To-Network message conveying no DNETs is received.

**Notes to Tester: This test is to be run after test 10.2.2.2.5 such that R2-3 is known to be a router to network 6.**

Test Steps:

1. TRANSMIT PORT B,  
DESTINATION = LOCAL BROADCAST,  
SOURCE = R2-3,  
Router-Busy-To-Network
2. RECEIVE PORT A,  
DESTINATION = LOCAL BROADCAST,  
SOURCE = IUT,  
Router-Busy-To-Network,  
Network Numbers = 3, 6 | 6, 3
3. TRANSMIT PORT B,  
DESTINATION = LOCAL BROADCAST,  
SOURCE = R2-3,  
Router-Available-To-Network
4. RECEIVE PORT A,  
DESTINATION = LOCAL BROADCAST,  
SOURCE = IUT,  
Router-Available-To-Network,  
Network Numbers = 3, 6 | 6, 3
5. TRANSMIT PORT A,  
DA = IUT,  
SOURCE = D1A,  
DNET = 3,  
DADR = D3D,  
Hop Count = 255,  
ReadProperty-Request,  
'Object Identifier' = (any BACnet standard object),  
'Property Identifier' = (any required property of the specified object)
6. RECEIVE PORT B,  
DA = R2-3,  
SOURCE = IUT,  
**SNET = 1,**  
**SADR = DIA,**  
DNET = 3,  
DADR = D3D,  
Hop Count = (any integer x:  $0 < x < 255$ ),  
ReadProperty-Request,  
'Object Identifier' = (the object identifier used in step 5),  
'Property Identifier' = (the property identifier used in step 5)
7. TRANSMIT PORT A,  
DA = IUT,  
SOURCE = D1A,  
DNET = 6,

DADR = (any valid device address),  
Hop Count = 255,  
ReadProperty-Request,  
'Object Identifier' = (any BACnet standard object),  
'Property Identifier' = (any required property of the specified object)

8. RECEIVE PORT B,  
DA = R2-3,  
SOURCE = IUT,  
SNET = 1,  
SADR = D1A,  
DNET = 6,  
DADR = (the address used in step 6),  
Hop Count = (any integer x: 0 < x < 255),  
ReadProperty-Request,  
'Object Identifier' = (the object identifier used in step 7),  
'Property Identifier' = (the property identifier used in step 7)

### 10.2.6 Network Layer Priority

Reason for Change: Correct test to specify correct DINFO information.

Purpose: To verify that the IUT can process and forward messages with all network priorities.

Test Steps:

1. TRANSMIT PORT A,  
DA = IUT,  
SA = D1A,  
Priority = B'00',  
DNET = 2,  
DADR = D2C,  
Hop Count = 255,  
BACnet-Confirmed-Request-PDU,  
'Service Choice' = ReadProperty-Request,  
'Object Identifier' = (any object identifier),  
'Property Identifier' = (any property of the specified object)
2. RECEIVE PORT B,  
SADA = D2C,  
SA = IUT,  
Priority = B'00',  
SNET = 1,  
SDR = D1A,  
BACnet-Confirmed-Request-PDU,  
'Service Choice' = ReadProperty-Request,  
'Object Identifier' = (the object identifier used in step 1),  
'Property Identifier' = (the property identifier used in step 1)
3. TRANSMIT PORT A,  
DA = IUT,  
SA = D1A,  
Priority = B'01',  
DNET = 2,  
DADR = D2C,  
Hop Count = 255,  
BACnet-Confirmed-Request-PDU,  
'Service Choice' = ReadProperty-Request,  
'Object Identifier' = (any object identifier),  
'Property Identifier' = (any property of the specified object)
4. RECEIVE PORT B,  
DA = D2C,  
SA = IUT,  
Priority = B'01',

- SNET = 1,
  - SDR = D1A,
  - BACnet-Confirmed-Request-PDU,
  - 'Service Choice' = ReadProperty-Request,
  - 'Object Identifier' = (the object identifier used in step 3),
  - 'Property Identifier' = (the property identifier used in step 3)
5. TRANSMIT PORT A,
- DA = IUT,
  - SA = D1A,
  - Priority = B'10',
  - DNET = 2,
  - DADR = D2C,
  - Hop Count = 255,
  - BACnet-Confirmed-Request-PDU,
  - 'Service Choice' = ReadProperty-Request,
  - 'Object Identifier' = (any object identifier),
  - 'Property Identifier' = (any property of the specified object)
6. RECEIVE PORT B,
- DA = D2C,
  - SA = IUT,
  - Priority = B'10',
  - SNET = 1,
  - SDR = D1A,
  - BACnet-Confirmed-Request-PDU,
  - 'Service Choice' = ReadProperty-Request,
  - 'Object Identifier' = (the object identifier used in step 5),
  - 'Property Identifier' = (the property identifier used in step 5)
7. TRANSMIT PORT A,
- DA = IUT,
  - SA = D1A,
  - Priority = B'11',
  - DNET = 2,
  - DADR = D2C,
  - Hop Count = 255,
  - BACnet-Confirmed-Request-PDU,
  - 'Service Choice' = ReadProperty-Request,
  - 'Object Identifier' = (any object identifier),
  - 'Property Identifier' = (any property of the specified object)
8. RECEIVE PORT B,
- DA = D2C,
  - SA = IUT,
  - Priority = B'11',
  - SNET = 1,
  - SDR = D1A,
  - BACnet-Confirmed-Request-PDU,
  - 'Service Choice' = ReadProperty-Request,
  - 'Object Identifier' = (the object identifier used in step 7),
  - 'Property Identifier' = (the property identifier used in step 7)

#### 10.2.2.4.2 Forwarding Router-Busy-To-Network Information for all DNETs

Reason for Change: Add new Notes to Tester.

Purpose: To verify that the IUT correctly forwards information indicating that all DNETs reachable through a particular router are temporarily unreachable because of traffic congestion.

**Notes to Tester: This test is to be run after test 10.2.2.2.5 such that R2-3 is known to be a router to network 6.**

Test Steps:

1. TRANSMIT PORT B,  
DESTINATION = LOCAL BROADCAST,  
SOURCE = R2-3,  
Router-Busy-To-Network
2. RECEIVE PORT A,  
DESTINATION = LOCAL BROADCAST,  
SOURCE = IUT,  
Router-Busy-To-Network,  
Network Numbers = 3,6 | 6,3 | (absent)

#### 10.2.3.6.2 Successful Attempt to Locate Router

Reason for Change: Correct test to specify correct DA information.

Purpose: To verify that the IUT will attempt to locate a router to an unknown network. When successful it forwards the message to the next router on the path.

Configuration Requirements: The IUT shall be configured to know only about the directly-connected networks.

Tomin: vendor defined minimum time the router waits for a response to the Who-Is-Router-To-Network request.

Notes to Tester: The standard does not provide any guidance on how long a router should wait before declaring that the attempt to locate the next router failed. While there is no explicit minimum time, it is expected that routers wait long enough that the attempt would succeed if the next hop router responded immediately.

Test Steps:

1. TRANSMIT PORT A,  
DA = IUT,  
SA = R1-5,  
DNET = 3,  
DADR = D3D,  
SNET = 5,  
SADR = D5F,  
Hop Count = 254,  
BACnet-Confirmed-Request-PDU,  
'Service Choice' = ReadProperty-Request,  
'Object Identifier' = (any object identifier),  
'Property Identifier' = (any property of the specified object)
2. RECEIVE PORT B,  
DESTINATION = LOCAL BROADCAST, SOURCE = IUT, Who-Is-Router-To-Network,  
Network Number = 3
3. WAIT any time less than Tomin
4. TRANSMIT PORT B,  
DESTINATION = LOCAL BROADCAST, SOURCE = R2-3, I-Am-Router-To-Network,  
Network Numbers = 3
5. RECEIVE PORT B,  
~~SA DA~~ = R2-3,  
SA = IUT,  
DNET = 3,  
DADR = D3D,  
SNET = 5,  
SADR = D5F,  
Hop Count = (any integer x: 0 < x < 254), BACnet-Confirmed-Request-PDU,  
'Service Choice' = ReadProperty-Request,  
'Object Identifier' = (the object identifier used in step 1), 'Property Identifier' = (the property identifier used in step 1)

**BTL-23.3 cr1-13: Loosen Lifetime Restrictions on Subscribe COV Test [BTLWG-1460. CR-0557]**

**Overview:**

Lifetime in the Test Concept is too restrictive. See CR09557,

**Changes:**

---

**Checklist Changes**

---

None

---

**Test Plan Changes**

---

**4.10.1 Base Requirements**

...	
<b>135.1-2023BTL - 9.10.1.8 - Updating Existing Subscriptions</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	

---

**Specified Test Changes**

---

**9.10.1.8 Updating Existing Subscriptions**

Reason for Change: Lifetime was too restrictive.

Purpose: To verify that the IUT correctly responds to a SubscribeCOV request to update the lifetime of a subscription. 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 notifications is made for 60 seconds. Before that subscription has expired a second subscription is made for 300 seconds. When the notification is sent in response to the second subscription the lifetime is checked to verify that it is **approximately 300 seconds but not** greater than **60 but less than** 300 seconds.

Test Steps:

1. TRANSMIT SubscribeCOV-Request,
  - 'Subscriber Process Identifier' = (PID1, any valid process identifier),
  - 'Monitored Object Identifier' = (O1, any object supporting COV notifications),
  - 'Issue Confirmed Notifications' = TRUE | FALSE,
  - 'Lifetime' = 60
2. RECEIVE BACnet-SimpleACK-PDU
3. IF (the subscription was for confirmed notifications) THEN
  - BEFORE Notification Fail Time**
  - RECEIVE ConfirmedCOVNotification-Request,
    - 'Subscriber Process Identifier' = PID1,
    - 'Initiating Device Identifier' = IUT,
    - 'Monitored Object Identifier' = O1,
    - 'Time Remaining' = (~60, but not greater than 60 or less than 60),
    - 'List of Values' = (values appropriate to the object type of the monitored object)
  - TRANSMIT BACnet-SimpleACK-PDU
  - ELSE

**BEFORE Notification Fail Time**

RECEIVE UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = PID1,

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = O1,

'Time Remaining' = (~60, but not greater than 60),

'List of Values' = (values appropriate to the object type of the monitored object)

4. TRANSMIT SubscribeCOV-Request,

'Subscriber Process Identifier' = PID1,

'Monitored Object Identifier' = O1,

'Issue Confirmed Notifications' = TRUE | FALSE,

'Lifetime' = (300T1, a value between 180 and 300 seconds)

5. RECEIVE BACnet-SimpleACK-PDU

6. IF (the subscription was for confirmed notifications) THEN

**BEFORE Notification Fail Time**

RECEIVE ConfirmedCOVNotification-Request,

'Subscriber Process Identifier' = PID1,

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = O1,

'Time Remaining' = (~300, but not greater than 300-T1, but not greater than T1),

'List of Values' = (values appropriate to the object type of the monitored object)

TRANSMIT BACnet-SimpleACK-PDU

ELSE

**BEFORE Notification Fail Time**

RECEIVE UnconfirmedCOVNotification-Request,

'Subscriber Process Identifier' = PID1,

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = O1,

'Time Remaining' = (~300, but not greater than 300-T1, but not greater than T1),

'List of Values' = (values appropriate to the object type of the monitored object)