BACnet® TESTING LABORATORIES ADDENDA

Addendum Imp1 to BTL Test Package 26.1

Revision final
Revised 6/3/2026

Approved by the BTL Working Group on March 19, 2026;
Approved by the BTL Working Group Voting Members on June 22, 2026;
Published on July 3, 2026.

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

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

FOREWORD

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

BTL-26.1 imp1-1: T-ATR-A Should Include Testing of Algorithmic BUFFER_READY Notifications [BTLWG-1079].....2
BTL-26.1 imp1-2: Door_Unlock_Delay_Time Test [BTLWG-1651] 4
BTL-26.1 imp1-3: 7.3.2.4.1 Reinitializing the Samples [BTLWG-1652].....6
BTL-26.1 imp1-4: 10.8.4.3 Route Global Broadcast from Remote Device to Virtual Devices [BTLWG-1662] 8
BTL-26.1 imp1-5: 7.3.2.47.1.2 Timer Running then Expired Test [BTLWG-1666] 19
BTL-26.1 imp1-6: 7.3.2.41.10 Denied Access Disabled Credential Test [BTLWG-1677].....21
BTL-26.1 imp1-7: 7.3.2.40.1 Commandable Present_Value Test [BTLWG-1679].....22
BTL-26.1 imp1-8: 8.4.8.7 Mode Transitions Are Not Correctly Tested [BTLWG-1681] 24
BTL-26.1 imp1-9: Network Port Object - ReadPropertyMultiple with Unknown Instance Test [BTLWG-1695] 28
BTL-26.1 imp1-10: IPv6 Missing a Test When in BBMD Mode [BTLWG-1699] 31

In the following document, language to be added to existing clauses within the BTL Test Package 26.1 is indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are proposed to be added, plain type is used throughout.

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

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

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp1-1: T-ATR-A Should Include Testing of Algorithmic BUFFER_READY Notifications [BTLWG-1079]

Overview:

The current test plan assumes that notifications come from Trend Log objects and not from Event Enrollment objects. In order to ensure that clients properly handle notifications generated by Event Enrollment objects, the test plan should be extended to include testing of this option.

Changes:

Checklist Changes

None

Test Plan Changes

[Add Test Directives for Tests under the section 7.5.4, 7.5.5, 7.9.3 and 7.9.4]

7.5 Trending - Automated Trend Retrieval - A

7.5.4 Processes ConfirmedEventNotification of BUFFER_READY Events

The IUT can process ConfirmedEventNotification messages indicating an 'Event Type' of Buffer_Ready, and respond with a series of ReadRange requests that access records from the Trend Log referenced in the notifications.

135.1-2025 - 8.21.6 - Reading a Range of Items Using Any Valid Range in Response to ConfirmedEventNotifications of the Buffer_Ready Event Type		
	Test Conditionality	Must be executed
	Test Directives	~~This test shall be executed with a reference server device claiming Protocol_Revision 3 or higher.~~ *Repeat the test with 'Event Object Identifier' set to Trend Log object and Event Enrollment object monitoring a Trend Log object.*
	Testing Hints	

7.5.5 Processes UnconfirmedEventNotification of BUFFER_READY Events

The IUT can process UnconfirmedEventNotification messages indicating an 'Event Type' of Buffer_Ready, and respond with a series of ReadRange requests that access records from the Trend Log referenced in the notifications.

135.1-2025 - 8.21.7 - Reading a Range of Items Using Any Valid Range in Response to UnconfirmedEventNotifications of the Buffer_Ready Event Type		
	Test Conditionality	Must be executed
	Test Directives	~~This test shall be executed with a reference server device claiming Protocol_Revision 3 or higher.~~ *Repeat the test with 'Event Object Identifier' set to Trend Log object and Event Enrollment object monitoring a Trend Log object.*
	Testing Hints	

7.9 Trending - Automated Multiple Value Retrieval - A

7.9.3 Processes ConfirmedEventNotification of BUFFER_READY Events

The IUT can process ConfirmedEventNotification messages indicating an 'Event Type' of Buffer_Ready, and respond with a series of ReadRange requests that access records from the Trend Log Multiple referenced in the notifications.

© 2026 by BACnet International. All rights reserved.

135.1-2025 - 8.21.6 - Reading a Range of Items Using Any Valid Range in Response to ConfirmedEventNotifications of the Buffer_Ready Event Type		
	Test Conditionality	Must be executed
	Test Directives	*Repeat the test with 'Event Object Identifier' set to Trend Log Multiple object and Event Enrollment object monitoring a Trend Log Multiple object.*
	Testing Hints	

7.9.4 Processes UnconfirmedEventNotification of BUFFER_READY Events

The IUT can process UnconfirmedEventNotification messages indicating an 'Event Type' of Buffer_Ready, and respond with a series of ReadRange requests that access records from the Trend Log Multiple referenced in the notifications.

135.1-2025 - 8.21.7 - Reading a Range of Items Using Any Valid Range in Response to UnconfirmedEventNotifications of the Buffer_Ready Event Type		
	Test Conditionality	Must be executed
	Test Directives	*Repeat the test with 'Event Object Identifier' set to Trend Log Multiple object and Event Enrollment object monitoring a Trend Log Multiple object.*
	Testing Hints	

Specified Test Changes

None

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp1-2: Door_Unlock_Delay_Time Test [BTLWG-1651]

Overview:

BTL Specified Tests-26.0:
7.3.2.40.6 Door_Unlock_Delay_Time Test:

In step 3, just before "Door_Unlock_Delay_Time" expires, we check if the "Lock_Status" is "Locked" or "Unlocked." The question arises: how much time should be allocated for it ("Door_Unlock_Delay_Time")?

This verification process involves waiting. Prior to this step, in Step 2, there's a waiting time known as "Internal Processing Fail Time." Additionally, the check is a manual one, which inherently takes time.

To this step's implementation, it is recommended to highlight in the testing phase that "Door_Unlock_Delay_Time" should be sufficiently long to accommodate these waiting periods. Moreover, it might be beneficial to substitute "BEFORE" with "WHILE" in step 3, then the VERIFY and CHECK would be part of a loop.

It also applies to the step 12 (Test EXTENDED_PULSE_UNLOCK)

Changes:

Checklist Changes

None

Test Plan Changes

3.42.7 Supports Door_Unlock_Delay_Time Property

The IUT contains or can be made to contain a writable or read-only Door_Unlock_Delay_Time property

~~135.1-2025~~*BTL* - 7.3.2.40.6 - Door_Unlock_Delay_Time Test		
	Test Conditionality	Must be executed *if the Lock_Status property is supported.*
	Test Directives	
	Testing Hints	

Specified Test Changes

7.3.2.40.6 Door_Unlock_Delay_Time Test

Reason for Change: Simplified test by removing output checks and only running if Lock_Status property exists..

Purpose: To verify that when the Door_Unlock_Delay_Time property has a non-zero value, the output is delayed in unlocking when a PULSE_UNLOCK or EXTENDED_PULSE_UNLOCK is written to the Present_Value and not when UNLOCK is written.

Test Concept: When unlocking the door by writing PULSE_UNLOCK to the Present_Value of the Access Door object, it is verified that the door is still locked for the specified Door_Pulse_Time, then the door is unlocked. The same test is done for EXTENDED_PULSE_UNLOCK, but this time, it is verified that the door is still locked for the specified Door_Extended_Pulse_Time then the door is unlocked.

Configuration Requirements: ~~The IUT shall be configured with a door control output that can be observed during the test.~~ The Relinquish_Default shall have the value LOCK. All writes to the Present_Value shall be performed at a priority higher than any internal algorithms writing to this property. Door_Unlock_Delay_Time shall be set to a non-zero value which is sufficient to *capture the value of the Lock_Status*~~observe the delay and check the status of the lock~~. Out_Of_Service shall be set to

© 2026 by BACnet International. All rights reserved.

FALSE. Prior to the test the Present_Value shall have the value LOCK and the IUT is in a state that would cause the door to be locked.

Test Steps:

-- Test PULSE_UNLOCK
1. WRITE Present_Value = PULSE_UNLOCK
2. WAIT (**Internal Processing Fail Time**)
3. BEFORE Door_Unlock_Delay_Time
~~4. IF (Lock_Status is present) THEN~~
4. VERIFY Lock_Status = LOCKED
~~6. CHECK (that the door control output is in a state that would cause the door to be locked)~~
~~*5. WAIT (Door_Unlock_Delay_Time)*~~
~~7. IF (Lock_Status is present) THEN~~
6. VERIFY Lock_Status = UNLOCKED
~~9. CHECK (that the door control output is in a state that would cause the door to be unlocked)~~
7. WAIT (Door_Pulse_Time)
10. VERIFY Present_Value = LOCK
~~11. IF (Lock_Status is present) THEN~~
8. VERIFY Lock_Status = LOCKED
~~14. CHECK (that the door control output is in a state that would cause the door to be locked)~~

-- Test EXTENDED_PULSE_UNLOCK
9. WRITE Present_Value = EXTENDED_PULSE_UNLOCK
10. WAIT (**Internal Processing Fail Time**)
11. BEFORE Door_Unlock_Delay_Time
~~16. IF (Lock_Status is present) THEN~~
12. VERIFY Lock_Status = LOCKED
~~20. CHECK (that the door control output is in a state that would cause the door to be locked)~~
~~*13. WAIT (Door_Unlock_Delay_Time)*~~
~~19. IF (Lock_Status is present) THEN~~
14. VERIFY Lock_Status = UNLOCKED
~~23. CHECK (that the door control output is in a state that would cause the door to be unlocked)~~
15. WAIT (Door_Extended_Pulse_Time)
16. VERIFY Present_Value = LOCK
~~23. IF (Lock_Status is present) THEN~~
17 VERIFY Lock_Status = LOCKED
~~28. CHECK (that the door control output is in a state that would cause the door to be locked)~~

-- Test UNLOCK
18. WRITE Present_Value = UNLOCK
19. WAIT (**Internal Processing Fail Time**)
~~28. IF (Lock_Status is present) THEN~~
20. VERIFY Lock_Status = UNLOCKED
~~33. CHECK (that the door control output is in a state that would cause the door to be unlocked)~~

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp1-3: 7.3.2.4.1 Reinitializing the Samples [BTLWG-1652]

Overview:

The test in ASHRAE 135.1-2025 is failing its purpose without verifying the appropriate properties for their initial values.

Proposed Changes:
Verifying the initial values.

Changes:

Checklist Changes

None

Test Plan Changes

[Change all references to test 7.3.2.4.1 from 135.1-2025 to BTL]

Specified Test Changes

7.3.2.4.1 Reinitializing the Samples

Reason for Change: Add verify for initial values.

Purpose: To verify that an Averaging object correctly resets the Attempted_Samples, Valid_Samples, Minimum_Value, Average_Value, and Maximum_Value when Attempted_Samples, Object_Property_Reference, Window_Interval, or Window_Samples are changed.

Test Concept: The IUT is configured with an Averaging object that is actively monitoring some property value. The sampling is reinitialized by writing to the Attempted_Samples, Window_Interval, Window_Samples and Object_Property_Reference in turn. After each reinitialization verify that new sampling has begun.

Configuration Requirements: The IUT shall be configured with an Averaging object that is actively monitoring some property value. The sampling interval shall be long enough to permit the TD to verify that the sample is properly reinitialized.

Test Steps:
1. WRITE Attempted_Samples = 0,
2. VERIFY Attempted_Samples = 0,
3. VERIFY Minimum_Value = INF,
4. VERIFY Maximum_Value = -INF,
5. VERIFY Average_Value = NaN,
6. VERIFY Valid_Samples = 0
7.~~2.~~ WAIT (at least two sample times),
8.~~3.~~ VERIFY Minimum_Value = (a value x: -INF < x < INF),
9.~~4.~~ VERIFY Average_Value = (a value ≠ NaN),
10.~~5.~~ VERIFY Maximum_Value = (a value x: Minimum_Value <= x < INF),
11.~~6.~~ VERIFY Attempted_Samples = (a value x >= 2),
12.~~7.~~ VERIFY Valid_Samples = (a value x >= 2),
13.~~8.~~ WRITE Window_Interval = (any new value that will result in an appropriate sample time),
14. VERIFY Attempted_Samples = 0,
15. VERIFY Minimum_Value = INF,
16. VERIFY Maximum_Value = -INF,
17. VERIFY Average_Value = NaN,

© 2026 by BACnet International. All rights reserved.

18. VERIFY Valid_Samples = 0
19. ~~9.~~ WAIT (at least two sample times),
20. ~~10.~~ VERIFY Minimum_Value = (a value x: -INF < x < INF),
21. ~~11.~~ VERIFY Average_Value = (a value ≠ NaN),
22. ~~12.~~ VERIFY Maximum_Value = (a value x: Minimum_Value <= x < INF),
23. ~~13.~~ VERIFY Attempted_Samples = (a value x >= 2),
24. ~~14.~~ VERIFY Valid_Samples = (a value x >= 2),
25. ~~15.~~ WRITE Window_Samples = (any new value that will result in an appropriate sample time),
26. VERIFY Attempted Samples = 0,
27. VERIFY Minimum_Value = INF,
28. VERIFY Maximum_Value = -INF,
29. VERIFY Average Value = NaN,
30. VERIFY Valid Samples = 0
31. WAIT (at least two sample times),
32. VERIFY Minimum Value = (a value x: -INF < x < INF),
33. VERIFY Average_Value = (a value ≠ NaN),
34. VERIFY Maximum_Value = (a value x: Minimum_Value <= x < INF),
35. VERIFY Attempted Samples = (a value x >= 2),
36. VERIFY Valid Samples = (a value x >= 2),
37. WRITE Window_Samples = (any new value that will result in an appropriate sample time),

38. ~~16.~~ IF (Object_Property_Reference is writable) THEN {
WAIT (at least two sample times),
VERIFY Minimum_Value = (a value x: -INF < x < INF),
VERIFY Average_Value = (a value ≠ NaN),
VERIFY Maximum_Value = (a value x: Minimum_Value <= x < INF),
VERIFY Attempted_Samples = (a value x >= 2),
VERIFY Valid_Samples = (a value x >= 2),
WRITE Object_Property_Reference = (any new value),
IF (Samples_are_taken_immediately) THEN
VERIFY Attempted_Samples = 1,
VERIFY Minimum_Value = Average_Value,,
VERIFY Maximum_Value = Average_Value,
VERIFY Valid_Samples = 1
ELSE
VERIFY Attempted_Samples = 0,
VERIFY Minimum_Value = INF,
VERIFY Maximum_Value = -INF,
VERIFY Average_Value = NaN,
VERIFY Valid_Samples = 0
}

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp1-4: 10.8.4.3 Route Global Broadcast from Remote Device to Virtual Devices [BTLWG-1662]

Overview:

In this test, three test options are mentioned: who-is, who-has, and generic.

In the test, a global broadcast is sent to the IUT. In addition to a response as a global broadcast, other/optional responses from the IUT are indicated in parentheses.

For the I-am response to the who-is, the last-mentioned option includes, among others:
DADR = (the MAC address of D3D).

This is not the case for who-has.

Changes:

Checklist Changes

None

Test Plan Changes

10.1.3 Routes Packets Between a Physical LAN and One or More Virtual LANs

...		
***BTL~~135.1-2025~~* - 10.8.4.1 - Broadcasts that Should Be Ignored**		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
***BTL~~135.1-2025~~* - 10.8.4.2 - Route Global Broadcast from a Local Device to Virtual Devices**		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
***BTL~~135.1-2025~~* - 10.8.4.3 - Route Global Broadcast from a Remote Device to Virtual Devices**		
	Test Conditionality	Must be executed.
	Test Directives	*Repeat this test with SADR of length = 1 and again with length = 6.*~~Use varying the lengths of SADR (1-6 bytes).~~
	Testing Hints	
***BTL~~135.1-2025~~* - 10.8.4.4 - Route Remote Broadcast from a Local Device to Virtual Devices**		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
***BTL~~135.1-2025~~* - 10.8.4.5 - Route Remote Broadcast from a Remote Device to Virtual Devices**		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
***BTL~~135.1-2025~~* - 10.8.4.6 - Route Global Broadcast Message from a Virtual Device**		
	Test Conditionality	~~Must be executed.~~ *This test shall be skipped if no virtual device can transmit a global broadcast message.*
	Test Directives	
	Testing Hints	
...		

© 2026 by BACnet International. All rights reserved.

BTL~~135.1-2025~~ **- 10.8.4.8 - Route Remote Broadcast Message from a Virtual Device to a Remote Network**		
	Test Conditionality	~~Must be executed.~~ *This test shall be skipped if no virtual device can transmit a remote broadcast message to a remote network.*
	Test Directives	
	Testing Hints	
...		

Specified Test Changes

10.8.4 Routing of Broadcast APDUs to Virtual Devices

10.8.4.1 Broadcasts that Should Be Ignored

Reason for Change: Removed optional tests.

Purpose: To verify that the IUT will not route APDUs which are locally broadcast on the directly connected physical BACnet network or remotely broadcast to a network that is reachable through the same port the message was received from.

Test Concept: *Initiate a local broadcast Who-Is message to any virtual device, VD1A, on the virtual network and check the IUT does not initiate an I-Am message for VD1A. Repeat the test with a directed Who-Is from a remote network and again with a local broadcast Who-Is message destined for a remote network.* ~~In order to verify that a broadcast message is not routed, we need to look for some indication that the message was routed. Two commonly supported services that can be used for this test are Who-Is and Who-Has, but complications may arise because a device is allowed to transmit the expected responses (I-Am and I-Have, respectively) at any time. There are a few other services that may also be used, but special device configuration will most likely be required. The tester shall choose one of the following test options, with the requirement that the option chosen must use services supported by at least one of the IUT's virtual devices.~~

~~This test shall be skipped if the IUT's virtual devices are not capable of executing any application services that may be broadcast.~~

~~A. Who-Is Option~~

~~This test option is an alternative for IUTs having a virtual device that supports the execution of the Who-Is service. This test shall be run after all tests for the execution of the Who-Is service have been run with a passing result. One virtual device that supports the execution of the Who-Is service shall be selected for this test.~~

Test Steps:

1. TRANSMIT
   DA = LOCAL BROADCAST,
   SA = TD,
   Who-Is-Request,
   'Device Instance Range Low Limit' = (~~the Device object instance number of the virtual device~~*VD1A*),
   'Device Instance Range High Limit' = (~~the Device object instance number of the virtual device~~*VD1A*)
2. WAIT **Internal Processing Fail Time**
3. CHECK (verify that the IUT does not transmit an I-Am message *for VD1A*~~from the virtual device selected in step 1~~)
4. TRANSMIT
   DA = LOCAL BROADCAST,
   SA = R2-3,
   SNET = 3,
   SADR = D3D,
   Who-Is-Request,
   'Device Instance Range Low Limit' = (~~the Device object instance number of the virtual device~~*VD1A*)
   'Device Instance Range High Limit' = (~~the Device object instance number of the virtual device~~*VD1A*)
5. WAIT **Internal Processing Fail Time**
6. CHECK (verify that the IUT does not transmit an I-Am message *for VD1A*~~from the virtual device selected in step 4~~)
7. TRANSMIT
   DA = LOCAL BROADCAST,

© 2026 by BACnet International. All rights reserved.

SA = TD,
DNET = 3,
DLEN = 0,
Hop Count = 255,
Who-Is-Request,
'Device Instance Range Low Limit' = (~~the Device object instance number of the virtual device~~*VD1A*)
'Device Instance Range High Limit' = (~~the Device object instance number of the virtual device~~*VD1A*)

8. WAIT **Internal Processing Fail Time**
9. CHECK (verify that the IUT does not transmit an I-Am message *for VD1A*~~from the virtual device selected in step 7~~)

~~Notes to Tester: I-Am messages may be sent at any time by any device. If one or both of the CHECK steps fail, then repeat the test until you can determine with confidence whether the I-Am messages are in response to the Who-Is (a failing result) or are being transmitted for some other reason (a passing result).~~

~~B. Who-Has Option~~

~~This test option is an alternative for IUTs that have a virtual device that supports the execution of the Who-Has service. This test shall be run after all tests for the execution of the Who-Has service have been run with a passing result. One virtual device that supports the execution of the Who-Has service shall be selected for this test.~~

~~Test Steps:~~

~~1. TRANSMIT~~
~~DA = LOCAL BROADCAST,~~
~~SA = TD,~~
~~Who-Has-Request,~~
~~'Device Instance Range Low Limit' = (the Device object instance number of the virtual device),~~
~~'Device Instance Range High Limit' = (the Device object instance number of the virtual device),~~
~~'Object Identifier' = (any object identifier of an object in the virtual device)~~
~~2. WAIT **Internal Processing Fail Time**~~
~~3. CHECK (verify that the IUT does not transmit an I-Have message from the virtual device selected in step 1 that contains the object identifier used in step 1)~~
~~4. TRANSMIT~~
~~DA = LOCAL BROADCAST,~~
~~SA = R2-3,~~
~~SNET = 3,~~
~~SADR = D3D,~~
~~Who-Has-Request,~~
~~'Device Instance Range Low Limit' = (the Device object instance number of the virtual device),~~
~~'Device Instance Range High Limit' = (the Device object instance number of the virtual device),~~
~~'Object Identifier' = (any object identifier of an object in the virtual device)~~
~~5. WAIT **Internal Processing Fail Time**~~
~~6. CHECK (verify that the IUT does not transmit an I-Have message from the virtual device selected in step 4 that contains the object identifier used in step 4)~~
~~7. TRANSMIT~~
~~DA = LOCAL BROADCAST,~~
~~SA = TD,~~
~~DNET = 3,~~
~~DLEN = 0,~~
~~Hop Count = 255,~~
~~Who-Has-Request,~~
~~'Device Instance Range Low Limit' = (the Device object instance number of the virtual device),~~
~~'Device Instance Range High Limit' = (the Device object instance number of the virtual device),~~
~~'Object Identifier' = (any object identifier of an object in the virtual device)~~
~~8. WAIT **Internal Processing Fail Time**~~
~~9. CHECK (verify that the IUT does not transmit an I-Have message from the virtual device selected in step 4 that contains the object identifier used in step 7)~~

© 2026 by BACnet International. All rights reserved.

~~Notes to Tester: I-Have messages may be sent at any time by any device. If one or both of the CHECK steps fail, then repeat the test until you can determine with confidence whether the I-Have messages are in response to the Who-Has (a failing result) or are being transmitted for some other reason (a passing result).~~

~~C. Generic Test Option~~

~~This test option is intended for IUTs whose virtual devices do not support the execution of the Who-Has or Who-Is services. This option uses a service that may be broadcast and that is supported by one of the IUT's virtual devices. This test shall be run after all tests for the execution of the particular service have been run with a passing result.~~

~~Configuration requirements: Configure the IUT so that the receipt of a particular service request by a particular virtual device will cause some indication that is visible to the tester. Verify that the indication occurs if the service request is sent directly to the virtual device or remote broadcast on Network 1.~~

~~Test Steps:~~

~~1. TRANSMIT~~
~~DA = LOCAL BROADCAST,~~
~~SA = TD,~~
~~BACnet-Unconfirmed-Request-PDU~~
~~2. WAIT **Internal Processing Fail Time**~~
~~3. CHECK (verify that the indication does not occur)~~
~~4. TRANSMIT~~
~~DA = LOCAL BROADCAST,~~
~~SA = R2-3,~~
~~SNET = 3,~~
~~SADR = D3D,~~
~~BACnet-Unconfirmed-Service-Request~~
~~5. WAIT **Internal Processing Fail Time**~~
~~6. CHECK (verify that the indication does not occur)~~
~~7. TRANSMIT~~
~~DA = LOCAL BROADCAST,~~
~~SA = TD,~~
~~DNET = 3,~~
~~DLEN = 0,~~
~~Hop Count = 255,~~
~~BACnet-Unconfirmed-Service-Request~~
~~8. WAIT **Internal Processing Fail Time**~~
~~9. CHECK (verify that the indication does not occur)~~

10.8.4.2 Route Global Broadcast from a Local Device to Virtual Devices

Reason for Change: Removed optional tests.

Purpose: To verify that the IUT properly forwards global broadcast messages originating on a local network to its virtual devices. ~~This test shall be skipped if the IUT's virtual devices are not capable of executing any application services that may be broadcast.~~

Test Concept: *Initiate a global broadcast Who-Has message to any virtual device, VD1A, on the virtual network and check the IUT initiate an I-Have message for VD1A.* ~~A broadcast message of a type that the virtual device will execute is sent to the IUT. It is then verified that a correct response is generated.~~

~~The tester selects one of the following test options, depending on which services are supported by the virtual devices in the IUT.~~

~~A. Who-Is Test Option~~

~~Select one virtual device in the IUT that supports the execution of Who-Is requests.~~

~~Test Steps:~~

© 2026 by BACnet International. All rights reserved.

~~1. TRANSMIT~~
~~DA = LOCAL BROADCAST,~~
~~SA = D2C,~~
~~DNET = GLOBAL BROADCAST,~~
~~DLEN = 0,~~
~~Hop Count = 255,~~
~~Who-Is-Request,~~
~~'Device Instance Range Low Limit' = (the Device object instance number of the virtual device),~~
~~'Device Instance Range High Limit' = (the Device object instance number of the virtual device)~~
~~2. RECEIVE~~
~~DA = LOCAL BROADCAST,~~
~~SA = IUT,~~
~~DNET = GLOBAL BROADCAST,~~
~~DLEN = 0,~~
~~SNET = 1,~~
~~SADR = (the MAC address of the virtual device),~~
~~Hop Count = (any integer x: 1 < x < 255),~~
~~I-Am-Request,~~
~~'I-Am Device Identifier' = (the Device object instance number of the virtual device),~~
~~'Max-APDU-Length-Accepted' = (any valid value),~~
~~'Segmentation Supported' = (any valid value),~~
~~'Vendor Identifier' = (any valid value)~~
~~(DA = LOCAL BROADCAST | D2C,~~
~~SA = IUT,~~
~~SNET = 1,~~
~~SADR = (the MAC address of the virtual device),~~
~~I-Am-Request,~~
~~'I-Am Device Identifier' = (the Device object instance number of the virtual device),~~
~~'Max-APDU-Length-Accepted' = (any valid value),~~
~~'Segmentation Supported' = (any valid value),~~
~~'Vendor Identifier' = (any valid value))~~

~~B. Who-Has Test Option~~

~~Select one virtual device in the IUT that supports the execution of Who-Has requests.~~

Test Steps:

1. TRANSMIT
   DA = LOCAL BROADCAST,
   SA = TD,
   DNET = GLOBAL BROADCAST,
   DLEN = 0,
   Hop Count = 255,
   Who-Has-Request,
   'Device Instance Range Low Limit' = (~~the Device object instance number of the virtual device~~*VD1A*)
   'Device Instance Range High Limit' = (~~the Device object instance number of the virtual device~~*VD1A*)
   'Object Identifier' = (*O1,* any object identifier of an object in the virtual device)
2. RECEIVE
   DA = LOCAL BROADCAST,
   SA = IUT,
   DNET = GLOBAL BROADCAST,
   DLEN = 0,
   SNET = 1,
   SADR = (the MAC address of ~~the virtual device~~*VD1A*),
   Hop Count = (any integer x: 1 < x < 255),
   I-Have-Request,
   'Device Identifier' = (~~the Device object instance number of the virtual device~~*VD1A*),
   'Object Identifier' = (~~the object identifier specified in step 1~~*O1*),
   ~~'Object Name' = (any valid value)~~

© 2026 by BACnet International. All rights reserved.

'Object Name' = (any valid value) |
(DA = LOCAL BROADCAST,
SA = IUT,
SNET = 1,
SADR = (the MAC address of ~~the virtual device~~*VD1A*),
I-Have-Request,
'Device Identifier' = (~~the Device object instance number of the virtual device~~*VD1A*),
'Object Identifier' = (~~the object identifier specified in step 1~~*O1*),
'Object Name' = (any valid value))

~~C. Generic Test Option~~

~~This test option is intended for IUTs whose virtual devices do not support the execution of the Who-Has or Who-Is services. This option uses a service that may be broadcast and that is supported by one of the IUT's virtual devices. This test shall be run after all tests for the execution of the particular service have been run with a passing result.~~

~~Configuration requirements: Configure the IUT so that the receipt of a particular service request by a particular virtual device will cause some indication that is visible to the tester.~~

~~Test Steps:~~

~~1. TRANSMIT~~
~~DA = LOCAL BROADCAST,~~
~~SA = TD,~~
~~DNET = GLOBAL BROADCAST,~~
~~DLEN = 0,~~
~~Hop Count = 255,~~
~~BACnet-Unconfirmed-Request-PDU~~
~~2. CHECK (verify that the indication occurs)~~

10.8.4.3 Route Global Broadcast from a Remote Device to Virtual Devices

Reason for Change: Removed optional tests.

Purpose: To verify that the IUT properly forwards global broadcast messages that originate on a remote network to its virtual devices. ~~This test shall be skipped if the IUT's virtual devices are not capable of executing any application services that may be broadcast.~~

Test Concept: Initiate a global broadcast Who-Is message from a remote device to any virtual device, VD1A, on the virtual network and check the IUT initiate an I-Am message for VD1A.
~~The tester shall select one of the following test options, depending on what services are supported by the virtual devices in the IUT.~~

~~A. Who-Is Test Option~~

~~Select one virtual device in the IUT that supports the execution of Who-Is requests.~~

Test Steps:

1. TRANSMIT
   DA = LOCAL BROADCAST,
   SA = R2-3,
   DNET = GLOBAL BROADCAST,
   DLEN = 0,
   SNET = 3,
   SADR = D3D,
   Hop Count = 254,
   Who-Is-Request,
   'Device Instance Range Low Limit' = (~~the Device object instance number of the virtual device~~*VD1A*)
   'Device Instance Range High Limit' = (~~the Device object instance number of the virtual device~~*VD1A*)
2. RECEIVE

© 2026 by BACnet International. All rights reserved.

DA = LOCAL BROADCAST,
SA = IUT,
DNET = GLOBAL BROADCAST,
DLEN = 0,
SNET = 1,
SADR = (the MAC address of ~~the virtual device~~*VD1A*),
Hop Count = (any integer x: 1 < x < 255),
I-Am-Request,
'I Am Device Identifier' = (~~the Device object instance number of the virtual device~~*VD1A*),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
~~'Vendor Identifier' = (any valid value)~~
'Vendor Identifier' = (any valid value) |
(DA = R2-3,
SA = IUT,
DNET = 3,
DLEN = 0,
SNET = 1,
SADR = (the MAC address of ~~the virtual device~~*VD1A*),
Hop Count = (any integer x: 1 < x < 255),
I-Am-Request,
'I Am Device Identifier' = (~~the Device object instance number of the virtual device~~*VD1A*),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
~~'Vendor Identifier' = (any valid value))~~
'Vendor Identifier' = (any valid value) |
(DA = R2-3,
SA = IUT,
DNET = 3,
DADR = (the MAC address of D3D),
SNET = 1,
SADR = (the MAC address of ~~the virtual device~~*VD1A*),
Hop Count = (any integer x: 1 < x < 255),
I-Am-Request,
'I Am Device Identifier' = (~~the Device object instance number of the virtual device~~*VD1A*),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (any valid value))

~~B. Who-Has Test Option~~

~~Select one virtual device in the IUT that supports the execution of Who-Has requests.~~

~~Test Steps:~~

~~1. TRANSMIT~~
~~DA = LOCAL BROADCAST,~~
~~SA = R2-3,~~
~~DNET = GLOBAL BROADCAST,~~
~~DLEN = 0,~~
~~SNET = 3,~~
~~SADR = D3D,~~
~~Hop Count = 254,~~
~~Who-Has-Request,~~
~~'Device Instance Range Low Limit' = (the Device object instance number of the virtual device),~~
~~'Device Instance Range High Limit' = (the Device object instance number of the virtual device),~~
~~'Object Identifier' = (any object identifier of an object in the virtual device)~~
~~2. RECEIVE~~
~~DA = LOCAL BROADCAST,~~
~~SA = IUT,~~

© 2026 by BACnet International. All rights reserved.

~~DNET = GLOBAL BROADCAST,~~
~~DLEN = 0,~~
~~SNET = 1,~~
~~SADR = (the MAC address of the virtual device),~~
~~Hop Count = (any integer x: 1 < x < 255),~~
~~I-Have-Request,~~
~~'Device Identifier' = (the Device object instance number of the virtual device),~~
~~'Object Identifier' = (the object identifier specified in step 1),~~
~~'Object Name' = (any valid value)~~
~~(DA = R2-3,~~
~~SA = IUT,~~
~~DNET = 3,~~
~~DLEN = 0,~~
~~SNET = 1,~~
~~SADR = (the MAC address of the virtual device),~~
~~Hop Count = (any integer x: 1 < x < 255),~~
~~I-Have-Request,~~
~~'Device Identifier' = (the Device object instance number of the virtual device),~~
~~'Object Identifier' = (the object identifier specified in step 1),~~
~~'Object Name' = (any valid value))~~

~~C. Generic Test Option~~

~~This test option is intended for IUTs whose virtual devices do not support the execution of the Who-Has or Who-Is services. This option uses a service that may be broadcast and that is supported by one of the IUT's virtual devices. This test shall be run after all tests for the execution of the particular service have been run with a passing result.~~

~~Configuration requirements: Configure the IUT so that the receipt of a particular service request by a particular virtual device will cause some indication that is visible to the tester.~~

~~Test Steps:~~

~~1. TRANSMIT~~
~~DA = LOCAL BROADCAST,~~
~~SA = R2-3,~~
~~DNET = GLOBAL BROADCAST,~~
~~DLEN = 0,~~
~~SNET = 3,~~
~~SADR = D3D,~~
~~Hop Count = 254,~~
~~BACnet-Unconfirmed-Request-PDU~~
~~2. CHECK (verify that the indication occurs)~~

10.8.4.4 Route Remote Broadcast from a Local Device to Virtual Devices

Reason for Change: Copied 10.8.4.2 Step 2 instead of referencing it.

Purpose: To verify that the IUT properly forwards remote broadcast messages originating on a local network to its virtual devices.

Test Concept: *Initiate a remote broadcast Who-Is message from a local device to any virtual device, VD1A, on the virtual network and check the IUT initiates an I-Am message for VD1A.*~~Execute test 10.8.4.2 (Route Global Broadcast from a Local Device to Virtual Devices) modified as follows. Instead of transmitting a global broadcast message, the TD shall transmit a remote broadcast message from a device on Network 2 directed to Network 1:~~

Test Steps:

1. TRANSMIT
   DA = LOCAL BROADCAST,
   SA = TD,
   DNET = 1,

© 2026 by BACnet International. All rights reserved.

DLEN = 0,
Hop Count = 255,
~~BACnet-Unconfirmed-Request-PDU~~
~~2. Continue with Step 2 from test 10.8.4.2~~
Who-Is-Request,
'Device Instance Range Low Limit' = (VD1A),
'Device Instance Range High Limit' = (VD1A)
2. RECEIVE
DA = LOCAL BROADCAST,
SA = IUT,
DNET = GLOBAL BROADCAST,
DLEN = 0,
SNET = 1,
SADR = (the MAC address of VD1A),
Hop Count = (any integer x: 1 < x < 255),
I-Am-Request,
'I Am Device Identifier' = (VD1A),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (any valid value) |
(DA = LOCAL BROADCAST,
SA = IUT,
SNET = 1,
SADR = (the MAC address of VD1A),
I-Am-Request,
'I Am Device Identifier' = (VD1A),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (any valid value))

~~This test shall be skipped if the IUT's virtual devices are not capable of executing any application services that may be broadcast.~~

10.8.4.5 Route Remote Broadcast from a Remote Device to Virtual Devices

Reason for Change: Copied 10.8.4.3 Step 2 instead of referencing it.

Purpose: To verify that the IUT properly forwards remote broadcast messages that originate on a remote network to its virtual devices.

Test Concept: *Initiate a remote broadcast Who-Is message from a remote device to any virtual device, VD1A, on the virtual network and check the IUT initiates an I-Am message for VD1A.*~~Execute test 10.8.4.3 (Route Global Broadcast from a Remote Device to Virtual Devices) modified as follows. Instead of transmitting a global broadcast message, the TD shall transmit a remote broadcast message from a device on Network 3 directed to Network 1:~~

Test Steps:

1. TRANSMIT
DA = IUT,
SA = R2-3,
DNET = 1,
DLEN = 0,
SNET = 3,
SADR = D3D,
Hop Count = 254,
~~BACnet-Unconfirmed-Request-PDU~~
~~2. Continue with Step 2 from test 10.8.4.3~~
Who-Is-Request,
'Device Instance Range Low Limit' = (VD1A)
'Device Instance Range High Limit' = (VD1A)
2. RECEIVE
DA = LOCAL BROADCAST,

© 2026 by BACnet International. All rights reserved.

SA = IUT,
DNET = GLOBAL BROADCAST,
DLEN = 0,
SNET = 1,
SADR = (the MAC address of VD1A),
Hop Count = (any integer x: 1 < x < 255),
I-Am-Request,
'I Am Device Identifier' = (VD1A),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (any valid value) |
(DA = R2-3,
SA = IUT,
DNET = 3,
DLEN = 0,
SNET = 1,
SADR = (the MAC address of VD1A),
Hop Count = (any integer x: 1 < x < 255),
I-Am-Request,
'I Am Device Identifier' = (VD1A),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (any valid value) |
(DA = R2-3,
SA = IUT,
DNET = 3,
DADR = (the MAC address of D3D),
SNET = 1,
SADR = (the MAC address of VD1A),
Hop Count = (any integer x: 1 < x < 255),
I-Am-Request,
'I Am Device Identifier' = (VD1A),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (any valid value))

~~This test shall be skipped if the IUT's virtual devices are not capable of executing any application services that may be broadcast.~~

10.8.4.6 Route Global Broadcast Message from a Virtual Device

Reason for Change: Fixed incorrect broadcast type.

Purpose: To verify that the IUT can route a global broadcast message from one of its virtual devices to the local (physical) network.

Test Concept: Make one of the virtual devices generate a *global*~~remote~~ broadcast, and verify that it is correctly formulated. This test shall be skipped if none of the IUT's virtual devices can transmit a global broadcast message.

Test Steps:

1. MAKE (the virtual device generate a *global*~~remote~~ broadcast message)
2. RECEIVE
   DA = LOCAL BROADCAST,
   SA = IUT,
   DNET = GLOBAL BROADCAST,
   DLEN = 0,
   SNET = 1,
   SADR = (MAC address of a virtual device on Network 1),
   Hop Count = (any integer x: 1 < x < 255),
   BACnet-Unconfirmed-Request-PDU

© 2026 by BACnet International. All rights reserved.

10.8.4.8 Route Remote Broadcast Message from a Virtual Device to a Remote Network

Reason for Change: Removed unnecessary Note to Tester.

Purpose: To verify that the IUT can route a remote broadcast message from a virtual device to a remote network.

Test Concept: The IUT shall be configured such that its routing table only contains entries for the directly connected networks (physical and virtual). One of the virtual devices is made to send a remote broadcast message to Network 3.

Configuration Requirements: The IUT shall be configured such that its routing table only contains entries for the directly connected networks (physical and virtual). This test shall be skipped if none of the IUT's virtual devices can issue a remote broadcast message.

Test Steps:

1. MAKE (the virtual device generate a remote broadcast message to ~~the~~ network 3)
2. RECEIVE
   DA = R2-3,
   SA = IUT,
   DNET = 3,
   DLEN = 0,
   SNET = 1,
   SADR = (MAC address of a virtual device on Network 1),
   Hop Count = (any integer x: 1 < x < 255),
   ~~BACnet-Unconfirmed-Request-PDU~~
   ~~| (DA = LOCAL_BROADCAST,~~
   BACnet-Unconfirmed-Request-PDU |
   (DA = LOCAL_BROADCAST,
   SA = IUT,
   DNET = 3,
   DLEN = 0,
   SNET = 1,
   SADR = (MAC address of a virtual device on Network 1),
   Hop Count = (any integer x: 1 < x < 255),
   BACnet-Unconfirmed-Request-PDU)

~~Notes to Tester: During the test, the TD shall answer any requests that the IUT generates while attempting to locate the route to network 3.~~

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp1-5: 7.3.2.47.1.2 Timer Running then Expired Test [BTLWG-1666]

Overview:

In step 12 it is difficult to verify the Update_Time against the “current date and time” because the CHECK in step 8 lasts an undefined time. The CHECK and the CHECK in step 3 can be completely removed as their intended purpose is checked with test “7.3.2.47.1.1 Timer State_Change_Values”

Changes:

Checklist Changes

None

Test Plan Changes

[move from 135.1 to BTL for all instances of 7.3.2.47.1.2]

Specified Test Changes

7.3.2.47.1.2 Timer Running then Expired Test

Reason for change: Changing test step order to improve verifying Update_Time against “the current date time” and remove unnecessary CHECKs.

Purpose: Verify that Timer T1 when set to RUNNING, enters the EXPIRED state after the configured time.

Test Concept: Start this test in the IDLE state. Write Timer_Running and observe it enter RUNNING state, with its current configuration. After time passes, observe it enters the EXPIRED state. Observe that specified properties take their required values.

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

Test Steps:

1. VERIFY Timer_State = IDLE
2. WRITE Timer_Running = TRUE
~~3. CHECK (IUT exhibits any changes configured in IDLE_TO_RUNNING transition)~~
~~4~~3. IF (Default_Timeout property is present in T1) THEN {
4. READ DV = Default_Timeout
5. VERIFY Initial_Timeout = DV
}
~~5~~6. VERIFY Timer_State = RUNNING
~~6~~7. READ RT = Present_Value
~~7~~8. WAIT RT
~~8. CHECK (IUT exhibits any changes configured in RUNNING_TO_EXPIRED transition)~~
9. VERIFY Timer_State = EXPIRED
~~10. VERIFY Present_Value = 0~~
~~11. VERIFY Last_State_Change = RUNNING_TO_EXPIRED~~
~~12~~*10.* IF (Update_Time property is present in T1) THEN {
11. READ UT = Update_Time
12. VERIFY UT ~= (the current date and time)
13. IF (Expiration_Time property is present in T1) THEN
14. VERIFY Expiration_Time = UT
}
ELSE

© 2026 by BACnet International. All rights reserved.

15. IF (Expiration_Time property is present in T1) THEN
16. VERIFY Expiration_Time ~= (the current date and time)
17. VERIFY Present_Value = 0
18. VERIFY Last_State_Change = RUNNING_TO_EXPIRED

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp1-6: 7.3.2.41.10 Denied Access Disabled Credential Test [BTLWG-1677]

Overview:

Added object references for all properties in the test steps to improve readability.

Changes:

Checklist Changes

None

Test Plan Changes

[move from 135.1 to BTL for all instances 7.3.2.41.10]

Specified Test Changes

7.3.2.41.10 Denied Access Disabled Credential Test

Reason for Change: Added object references for all properties in the test steps to improve readability.

Purpose: To test that a credential is denied access at an access point when the credential is disabled even when it has valid access rights.

Test Concept: A disabled credential is presented at an access point and access is denied. For testing purposes, the credential shall be disabled by setting the Expiration_Time to a time in the past.

Configuration Requirements: See Clause 7.3.2.41.1. This test requires the following additional configuration:

a) An access credential with valid access to AP1 shall be represented by Access Credential C1.
b) The Credential_Status property shall have the value ACTIVE.
c) The Reason_For_Disable property shall be empty.

Test Steps:

– test granted access when credential is active
1. VERIFY *C1,*Reason_For_Disable = ( )
2. VERIFY *C1,*Credential_Status = ACTIVE
3. MAKE (present credential C1 at credential reader for access point AP1)
4. VERIFY AP1,Access_Event = GRANTED
5. VERIFY AP1,Access_Event_Time = (the time that the credential C1 was presented)
6. VERIFY AP1,Access_Event_Credential = C1

– test denied access when credential is inactive
7. MAKE (*C1,*Expiration_Time = time < current time)
8. VERIFY *C1,*Reason_For_Disable = (DISABLED_EXPIRED)
9. VERIFY *C1,*Credential_Status = INACTIVE
10. MAKE (present credential C1 at credential reader for access point AP1)
11. VERIFY AP1,Access_Event = DENIED_CREDENTIAL_EXPIRED
12. VERIFY AP1,Access_Event_Time = (the time that the credential C1 was presented)
13. VERIFY AP1,Access_Event_Credential = C1

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp1-7: 7.3.2.40.1 Commandable Present_Value Test [BTLWG-1679]

Overview:

Missing Test Steps for "-- Test EXTENDED_PULSE_UNLOCK value".

Changes:

Checklist Changes

None

Test Plan Changes

[Change all references for test 7.3.2.40.1 from 135.1 to BTL]

Specified Test Changes

7.3.2.40.1 Commandable Present_Value Test

Reason for Change: Missing Test Steps for "-- Test EXTENDED_PULSE_UNLOCK value".

Purpose: To verify that writing to the Present_Value will cause a corresponding change to the physical output.

Test Concept: ~~The IUT shall be configured with a door control output that can be observed during the test.~~ The Present_Value property is written with each of the following values: UNLOCK, LOCK, PULSE_UNLOCK, EXTENDED_PULSE_UNLOCK, and the Access Door object is monitored to ensure that the door locks and unlocks appropriately.

Configuration Requirements: The Relinquish_Default shall have the value LOCK. All writes are at a priority higher than any internal algorithms writing to this property. Out_Of_Service shall be set to FALSE. Prior to the test the Present_Value shall have the value LOCK and the IUT is in a state that would cause the door to be locked.

Test Steps:

-- Test UNLOCK value
1. WRITE Present_Value = UNLOCK
2. WAIT (Internal Processing Fail Time)
3. IF (Lock_Status is present) THEN
4. VERIFY Lock_Status = UNLOCKED
~~4. CHECK (that the door control output is in a state that would cause the door to be unlocked)~~

-- Test LOCK value
5. WRITE Present_Value = LOCK
6. WAIT (Internal Processing Fail Time)
7. IF (Lock_Status is present) THEN
8. VERIFY Lock_Status = LOCKED
~~8. CHECK (that the door control output is in a state that would cause the door to be locked)~~

-- Test PULSE_UNLOCK value
9. WRITE Present_Value = PULSE_UNLOCK
10. WAIT (Internal Processing Fail Time + Door_Unlock_Delay_Time if present)
11. IF (Lock_Status is present) THEN
12. VERIFY Lock_Status = UNLOCKED
~~12. CHECK (that the IUT is in a state that would cause the door to be unlocked)~~
13. WAIT (Door_Pulse_Time)

© 2026 by BACnet International. All rights reserved.

14. VERIFY Present_Value = LOCK
15. IF (Lock_Status is present) THEN
16. VERIFY Lock_Status = LOCKED
~~16. CHECK (that the door control output is in a state that would cause the door to be locked)~~

-- Test EXTENDED_PULSE_UNLOCK value
17. WRITE Present_Value = EXTENDED_PULSE_UNLOCK
18. WAIT (Internal Processing Fail Time + Door_Unlock_Delay_Time if present)
19. IF (Lock_Status is present) THEN
20. VERIFY Lock_Status = UNLOCKED
21. WAIT (Door_Extended_Pulse_Time)
22. VERIFY Present_Value = LOCK
23. IF (Lock_Status is present) THEN
24. VERIFY Lock_Status = LOCKED

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp1-8: 8.4.8.7 Mode Transitions Are Not Correctly Tested [BTLWG-1681]

Overview:

8.4.8.7 assumes Mode requires a time delay which is incorrect. See 135-2024 Clause 13.3.8 c, g, and k.
The test transitions pMonitoredValue but does not include the actual event.

Changes:

Checklist Changes

None

Test Plan Changes

[In 5.22.10, move 135.1-2025 – 8.4.8.7 to BTL-8.4.8.7]

Specified Test Changes

8.4.8.7 Mode Transition Tests when Event State is Maintained

Reason for Change: Removed time delay for Mode.

Purpose: To verify the correct operation of the CHANGE_OF_LIFE_SAFETY event algorithm for objects transitioning between the NORMAL, OFFNORMAL and LIFE_SAFETY_ALARM event states when a mode change occurs. Tests are conducted when a mode change occurs, but the event state does not change. Tests are also conducted when a mode change occurs simultaneously with an event state change. In this latter case, the test verifies that the notification is immediate rather than waiting for the time delay.

Test Concept: The object begins the test in a NORMAL state. The Mode is changed. ~~After the time delay expires, the~~ *The* object should transmit an event notification message *immediately*. This operation is tested in the OFFNORMAL and LIFE_SAFETY_ALARM states as well.

~~The test is then repeated by changing the Mode property and simultaneously selecting a pMonitoredValue designated in pAlarmValues. The object should immediately enter the OFFNORMAL state and transmit an event notification message. pMonitoredValue is then changed to a value corresponding to a NORMAL state, and the Mode is simultaneously written. The object should immediately enter the NORMAL state and transmit an event notification message.~~

Configuration Requirements: The IUT shall be configured such that the Event_Enable property has a value of TRUE for the TO-OFFNORMAL, and TO-NORMAL transitions. The 'Issue_Confirmed_Notifications' parameter shall have a value of TRUE. The event-generating objects shall be in a NORMAL state at the start of the test.

Test Steps:

1. VERIFY Event_Detection_Enable = TRUE
2. VERIFY pCurrentState = NORMAL
~~2. CHECK (pCurrentState = NORMAL)~~
~~3. MAKE (pMonitoredValue have a value that corresponds to a NORMAL state)~~
3. IF (IUT supports another pMode value which maintains the NORMAL state) THEN {
4. WRITE pMode = (any valid value not equal to pMode)
~~MAKE (pMode = different value that maintains pCurrentState as NORMAL)~~
~~WAIT (pTimeDelayNormal)~~
5. BEFORE **Notification Fail Time**
6. RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,

© 2026 by BACnet International. All rights reserved.

'Event Object Identifier' = (the intrinsic reporting object or the Event Enrollment object being tested),
'Time Stamp' = (T1: any valid time stamp),
'Notification Class' = (the configured notification class),
'Priority' = (the value configured to correspond to a TO-NORMAL transition),
'Event Type' = CHANGE_OF_LIFE_SAFETY,
'Message Text' = (S1: optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = NORMAL,
'Event Values' = pMonitoredValue, pMode, pStatusFlags, pOperationExpected

7. TRANSMIT BACnet-SimpleACK-PDU
8. IF (Protocol_Revision is present AND Protocol_Revision >= 13) THEN
9. VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
10. VERIFY pCurrentState = NORMAL
~~IF (Protocol_Revision is present AND Protocol_Revision >= 1) THEN~~
11. VERIFY Event_Time_Stamps = (*, *, T1)
12. IF (Event_Message_Texts property exists) THEN
13. VERIFY Event_Message_Texts = (*, *, S1)

}

14. MAKE (pMonitoredValue have a value that corresponds to an OFFNORMAL state)
15. *WAIT (pTimeDelay)*
16. *BEFORE **Notification Fail Time***
17. *RECEIVE ConfirmedEventNotification-Request,*
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the intrinsic reporting object or the Event Enrollment object being tested),
'Time Stamp' = (T2: any valid time stamp),
'Notification Class' = (the configured notification class),
'Priority' = (the value configured to correspond to a TO-OFFNORMAL transition),
'Event Type' = CHANGE OF LIFE SAFETY,
'Message Text' = (S2, optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = OFFNORMAL,
'Event Values' = pMonitoredValue, pMode, pStatusFlags, pOperationExpected
18. *TRANSMIT BACnet-SimpleACK-PDU*
19. *IF (Protocol Revision is present AND Protocol Revision >= 13) THEN*
20. *VERIFY Status_Flags = (TRUE, FALSE, ?, ?)*
21. VERIFY pCurrentState = OFFNORMAL
22. *VERIFY Event Time Stamps = (T2, *, T1)*
23. *IF (Event_Message_Texts property exists) THEN*
24. *VERIFY Event_Message_Texts = (S2, *, S1)*
25. IF (IUT supports another pMode value which maintains the OFFNORMAL state) THEN {
26. *WRITE pMode = (any valid value not equal to pMode)*
~~MAKE (pMode = different value that maintains pCurrentState as OFFNORMAL)~~
~~WAIT (pTimeDelay)~~
27. BEFORE **Notification Fail Time**
28. RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the intrinsic reporting object or the Event Enrollment object being tested),
'Time Stamp' = (*T3*~~T2~~: any valid time stamp),
'Notification Class' = (the configured notification class),
'Priority' = (the value configured to correspond to a TO-OFFNORMAL transition),
'Event Type' = CHANGE_OF_LIFE_SAFETY,
'Message Text' = (*S3*~~S2:~~ optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,

© 2026 by BACnet International. All rights reserved.

'From State' = OFFNORMAL,
'To State' = OFFNORMAL,
'Event Values' = pMonitoredValue, pMode, pStatusFlags, pOperationExpected
29. TRANSMIT BACnet-SimpleACK-PDU
30. IF (Protocol_Revision is present AND Protocol_Revision >= 13) THEN
31. VERIFY Status_Flags = (TRUE, FALSE, ?, ?)
32. VERIFY pCurrentState = OFFNORMAL
~~IF (Protocol_Revision is present AND Protocol_Revision >= 1) THEN~~
33. VERIFY Event_Time_Stamps = (*T3*~~T2~~, *, *T1*~~*~~)
34. IF (Event_Message_Texts property exists) THEN
35. VERIFY Event_Message_Texts = (*S3*~~S2~~, *, ~~*~~*S1*)
}
36. MAKE (pMonitoredValue have a value that corresponds to a LIFE_SAFETY_ALARM state)
37. WAIT (pTimeDelay)
*38. BEFORE **Notification Fail Time***
39. RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the intrinsic reporting object or the Event Enrollment object being tested),
'Time Stamp' = (T4: any valid time stamp),
'Notification Class' = (the configured notification class),
'Priority' = (the value configured to correspond to a TO-OFFNORMAL transition),
'Event Type' = CHANGE_OF_LIFE_SAFETY,
'Message Text' = (S4, optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = OFFNORMAL,
'To State' = LIFE_SAFETY_ALARM,
'Event Values' = pMonitoredValue, pMode, pStatusFlags, pOperationExpected
40. TRANSMIT BACnet-SimpleACK-PDU
41. IF (Protocol Revision is present AND Protocol Revision >= 13) THEN
42. VERIFY Status_Flags = (TRUE, FALSE, ?, ?)
43. VERIFY pCurrentState = OFFNORMAL
*44. VERIFY Event_Time_Stamps = (T4, *, T1)*
45. IF (Event_Message_Texts property exists) THEN
*46. VERIFY Event_Message_Texts = (S4, *, S1)*
47. IF (IUT supports another pMode value which maintains the LIFE_SAFETY_ALARM state) THEN {
48. WRITE pMode = (any valid value not equal to pMode)
~~MAKE (pMode = different value that maintains pCurrentState = LIFE_SAFETY_ALARM)~~
~~WAIT (pTimeDelay)~~
49. BEFORE **Notification Fail Time**
50. RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the intrinsic reporting object or the Event Enrollment object being tested),
'Time Stamp' = (*T5*~~T3~~: any valid time stamp),
'Notification Class' = (the configured notification class),
'Priority' = (the value configured to correspond to a *TO-OFFNORMAL* ~~TO NORMAL~~ transition),
'Event Type' = CHANGE_OF_LIFE_SAFETY,
'Message Text' = (*S5*~~S3:~~ optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = LIFE_SAFETY_ALARM,
'To State' = LIFE_SAFETY_ALARM,
~~'From State' = OFFNORMAL,~~
~~'To State' = OFFNORMAL,~~
'Event Values' = pMonitoredValue, pMode, pStatusFlags, pOperationExpected
51. TRANSMIT BACnet-SimpleACK-PDU
52. IF (Protocol_Revision is present AND Protocol_Revision >= 13) THEN
53. VERIFY Status_Flags = (TRUE, FALSE, ?, ?)

© 2026 by BACnet International. All rights reserved.

54. VERIFY pCurrentState = OFFNORMAL
~~IF (Protocol_Revision is present AND Protocol_Revision >= 1) THEN~~
55. VERIFY Event_Time_Stamps = (*T5~~T3~~*, *, *T1~~*~~*)
56. IF (Event_Message_Texts property exists) THEN
57. VERIFY Event_Message_Texts = (*S5~~S3~~*, *, *~~*~~S1*)

}

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp1-9: Network Port Object - ReadPropertyMultiple with Unknown Instance Test [BTLWG-1695]

Overview:

The requirement to process an RPM request with NETWORK_PORT and 4194303 was added in PR 21.

Changes:

Checklist Changes

None

Test Plan Changes

9.1.8 Supports the Network Port Object

…		
135.1-2023 - 9.20.1.14 - ReadPropertyMultiple of the Network Port Object using the Unknown Instance		
	Test Conditionality	*If IUT claims Protocol_Revision less than 21, this test shall be skipped.* If the IUT does not support execution of ReadPropertyMultiple this test shall be skipped.
	Test Directives	
	Testing Hints	

9.2.2 Supports the Network Port Object

…		
135.1-2023 - 9.20.1.14 - ReadPropertyMultiple of the Network Port Object using the Unknown Instance		
	Test Conditionality	*If IUT claims Protocol_Revision less than 21, this test shall be skipped.* If the IUT does not support execution of ReadPropertyMultiple this test shall be skipped.
	Test Directives	
	Testing Hints	

9.3.5 Supports the Network Port Object

…		
135.1-2023 - 9.20.1.14 - ReadPropertyMultiple of the Network Port Object using the Unknown Instance		
	Test Conditionality	*If IUT claims Protocol_Revision less than 21, this test shall be skipped.* If the IUT does not support execution of ReadPropertyMultiple this test shall be skipped.
	Test Directives	
	Testing Hints	

9.4.2 Supports the Network Port Object

© 2026 by BACnet International. All rights reserved.

…		
135.1-2023 - 9.20.1.14 - ReadPropertyMultiple of the Network Port Object using the Unknown Instance		
	Test Conditionality	*If IUT claims Protocol_Revision less than 21, this test shall be skipped.* If the IUT does not support execution of ReadPropertyMultiple this test shall be skipped.
	Test Directives	
	Testing Hints	

9.5.2 Supports the Network Port Object

…		
135.1-2023 - 9.20.1.14 - ReadPropertyMultiple of the Network Port Object using the Unknown Instance		
	Test Conditionality	*If IUT claims Protocol_Revision less than 21, this test shall be skipped.* If the IUT does not support execution of ReadPropertyMultiple this test shall be skipped.
	Test Directives	
	Testing Hints	

9.6.2 Supports the Network Port Object

…		
135.1-2023 - 9.20.1.14 - ReadPropertyMultiple of the Network Port Object using the Unknown Instance		
	Test Conditionality	*If IUT claims Protocol_Revision less than 21, this test shall be skipped.* If the IUT does not support execution of ReadPropertyMultiple this test shall be skipped.
	Test Directives	
	Testing Hints	

9.7.2 Supports the Network Port Object

…		
135.1-2023 - 9.20.1.14 - ReadPropertyMultiple of the Network Port Object using the Unknown Instance		
	Test Conditionality	*If IUT claims Protocol_Revision less than 21, this test shall be skipped.* If the IUT does not support execution of ReadPropertyMultiple this test shall be skipped.
	Test Directives	
	Testing Hints	

9.8.5 Supports the Network Port Object

…		
135.1-2023 - 9.20.1.14 - ReadPropertyMultiple of the Network Port Object using the Unknown Instance		
	Test Conditionality	*If IUT claims Protocol_Revision less than 21, this test shall be skipped.* If the IUT does not support execution of ReadPropertyMultiple this test shall be skipped.

© 2026 by BACnet International. All rights reserved.

Test Directives	
Testing Hints	

Specified Test Changes

None

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp1-10: IPv6 Missing a Test When in BBMD Mode [BTLWG-1699]

Overview:

Test ‘135.1-2025 - 12.3.11.3 - Negative Foreign Device Registration when BBMD_Accept_FD_Registrations is FALSE’ is missing from the IPv6 testplan section 9.8.4. It is currently only used under testplan section 10.7.2 for NM-BBMDC-B, which is an IPv4-only BIBB.

Changes:

Checklist Changes

None

Test Plan Changes

9.8 Data Link Layer - IPv6

[Add test 12.3.11.3 to section 9.8.4 of Testplan]

9.8.4 Is Able to Operate in BBMD Mode

…		
135.1-2025 - 12.3.11.3 - Negative Foreign Device Registration when BBMD_Accept_FD_Registrations is FALSE		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

Specified Test Changes

None

© 2026 by BACnet International. All rights reserved.