BACnet® TESTING LABORATORIES ADDENDA

Addendum imp4 to BTL Test Package 26.1

Revision Final
Revised 9/14/2026

Approved by the BTL Working Group on August 20, 2026;
Approved by the BTL Working Group Voting Members on September 14, 2026;
Published on September 16, 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 imp4-1: Access Point Object Resizable Authentication Policies Test <doc title> [BTLWG-858] 2
BTL-26.1 imp4-2: DM-DDA - Subordinate Support [BTLWG-1388] 4
BTL-26.1 imp4-3: Update pTimeDelay and pTimeDelayNormal Tests [BTLWG-1826] 17
BTL-26.1 imp4-4: Dotted Decimal Host Name Test Updates [BTLWG-1860] 41
BTL-26.1 imp4-5: Change of State Tests Require Local Date and Time [BTLWG-1861] 42
BTL-26.1 imp4-6: Notification Class Transition Test [BTLWG-1859] 44

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 imp4-1: Access Point Object Resizable Authentication Policies Test [BTLWG-858]

Overview:

Add a test to verify that Authentication_Policy_List and Authentication_Policy_Names are the same length, including when resizing them

Changes:

Checklist Changes

[Modify the Access Point Object section]

Access Point Object		
	R	Base Requirements
	S	Supports configurable Out_Of_Service property
	O	Supports writable Active_Authentication_Policy property
	O	Supports writable Lockout property
	O	Supports writable Threat_Level property
	O	Supports Occupancy_Upper_Limit_Enforced and Occupancy_Count_Adjust properties
	O[1]	Supports configurable Reliability_Evaluation_Inhibit property
	O	Supports intrinsic reporting
	O	*Supports resizable Authentication Policy Properties*
[1] Protocol_Revision 13 or higher must be claimed. IUT shall not claim this functionality if the only method of generating a fault condition is by writing to the Reliability property.		

Test Plan Changes

3.44 Access Point Object

[Add to Access Point Object Section]

3.44.X1 Supports Resizable Authentication Policy Properties

The IUT supports resizable Number_Of_Authentication_Policies, Authentication_Policy_List, and Authentication_Policy_Names properties.

BTL - 7.3.2.41.X1 - Resizable Authentication Policy Properties Test		
	Test Conditionality	*Must be executed.*
	Test Directives	
	Testing Hints	

Specified Test Changes

[Add a new test to BTL Specified Tests]

7.3.2.41.X1 Resizable Authentication Policy Properties Test

Reason for Change: New Test.

© 2026 by BACnet International. All rights reserved.

Purpose: This test verifies that when the size of an authentication policy property is changed, the size of the other authentication properties changes to match.

Test Concept: Change the size of each authentication policy property and verify the size of the other properties also changes to match.

Configuration Requirements: None.

Test Steps:

1. READ N1 = Number_Of_Authentication_Policies
2. IF (Number_Of_Authentication_Policies is writable) THEN
3. WRITE Number_Of_Authentication_Policies = (N2, any valid value other than N1)
   ELSE
4. MAKE (Number_Of_Authentication_Policies = (N2, any valid value other than N1)
5. IF (Authentication_Policy_List is present) THEN
6. VERIFY Authentication_Policy_List = N2, ARRAY INDEX = 0
7. IF (Authentication_Policy_Names is present) THEN
8. VERIFY Authentication_Policy_Names = N2, ARRAY INDEX = 0
9. IF (Authentication_Policy_List is present and its size can be changed) THEN {
10. READ N1 = Authentication_Policy_List, ARRAY INDEX = 0
11. IF (The size of Authentication_Policy_List is writable) THEN
12. WRITE Authentication_Policy_List = (N2, any valid value other than N1), ARRAY INDEX = 0
    ELSE
13. MAKE (The size of Authentication_Policy_List = N2, (any valid value other than N1))
14. VERIFY Number_Of_Authentication_Policies = N2
15. IF (Authentication_Policy_Names is present) THEN
16. VERIFY Authentication_Policy_Names = N2, ARRAY INDEX = 0
    }

17. IF (Authentication_Policy_Names is present and its size can be changed) THEN {
18. READ N1 = Authentication_Policy_Names, ARRAY INDEX = 0
19. IF (The size of Authentication_Policy_Names is writable) THEN
20. WRITE Authentication_Policy_Names = (N2, any valid value other than N1), ARRAY INDEX = 0
    ELSE
21. MAKE (The size of Authentication_Policy_Names = (N2, any valid value other than N1))
22. VERIFY Number_Of_Authentication_Policies = N2
23. IF (Authentication_Policy_List is present) THEN
24. VERIFY Authentication_Policy_List = N2, ARRAY INDEX = 0
    }

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp4-2: DM-DDA - Subordinate Support [BTLWG-1388]

Overview:

A subordinate node cannot claim DM-DDA-B because the BIBB requires support for I-Am.
Cleaned up tests.
Added additional testable points.
Incorrect fail times in both DM-DDA-A and -B (From 1389)

Changes:

Checklist Changes

None

Test Plan Changes

8.31 Device Management - Dynamic Device Assignment - A

8.31.1 Base Requirements

Base requirements must be met by any IUT that can claim this BIBB.

~~135.1-2025~~*BTL* - 9.43.1 - Uses Who-Is to Configure Devices Supporting the Who-Am-I Service		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
...		

8.31.2 Can Configure Device Object Instance Number Using the You-Are Service

The IUT can configure another device's object instance number using the You-Are service.

~~135.1-2025~~*BTL* - 9.43.2 - Configures Other Device's Object Instance Number		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

8.31.4 Can Configure Both Device Object Instance Number and MAC Address Using the You-Are Service

The IUT can configure another device's object instance number and MAC address using the You-Are service.

~~135.1-2025~~*BTL* - 8.45.2 - Configures Other Device's Object Instance Number and MAC Address		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

© 2026 by BACnet International. All rights reserved.

8.32 Device Management - Dynamic Device Assignment - B

8.32.1 Base Requirements

Base requirements must be met by any IUT that can claim this BIBB.

Verify EPICS		
	Test Conditionality	Must be executed.
	Test Directives	Verify the Device Object has the Serial_Number property ~~required to initiate Who-Am-I and execute You-Are~~.
	Testing Hints	
~~135.1-2025~~*BTL* - 9.44.2.2 - Only Accepts Configuration When Received Parameters Match		
	Test Conditionality	*Must be executed.* ~~This test shall be skipped if the IUT is an MS/TP subordinate node.~~
	Test Directives	
	Testing Hints	
~~135.1-2025~~*BTL* - 8.44.1 - Responds to Who-Is With Who-Am-I While in the Unconfigured State		
	Test Conditionality	*Must be executed.* ~~If the IUT does not support having a MAC address but no configured Device object instance number, this test shall be skipped.~~
	Test Directives	
	Testing Hints	
~~135.1-2025~~*BTL* - 9.44.1.6 - Retains Configuration Through Restarts		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
~~135.1-2025~~*BTL* - 9.44.1.7 - Unconfigurable by You-Are		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

8.32.2 Has Configurable Device Object Instance Number Using the You-Are Service

The IUT can be configured with a Device object instance number using the You-Are service.

~~135.1-2025~~*BTL* - 9.44.1.1 - Configurable Device Object Instance Number		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
~~135.1-2025~~*BTL* - 9.44.1.4 - Device Object Instance Number is Configurable Even When MAC Address is Also Sent		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

8.32.3 Has Configurable MAC Address Using the You-Are Service

The IUT can be configured with a MAC address using the You-Are service.

~~135.1-2025~~*BTL* - 9.44.1.2 - Configurable MAC Address		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
~~135.1-2025*BTL* - 9.44.2.1 - Only Supports Execution of the You-Are Service While Unconfigured~~		
	~~Test Conditionality~~	~~Must be executed.~~
	~~Test Directives~~	

© 2026 by BACnet International. All rights reserved.

	~~**Testing Hints**~~	
~~**135.1-2025*BTL***~~ **- 9.44.1.5 - MAC Address is Reconfigurable**		
	Test Conditionality	*Must be executed.* ~~If the IUT cannot be configured with a MAC address that can be changed using the You-Are service, this test shall be skipped.~~
	Test Directives	
	Testing Hints	
~~**135.1-2025*BTL***~~ **- 9.44.1.3 - Configurable Device Object Instance Number and MAC Address**		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
~~**135.1-2025 - 9.44.1.4 - Device Object Instance Number is Configurable Even When MAC Address is Also Sent**~~		
	~~**Test Conditionality**~~	~~If the IUT cannot be configured with a MAC address that cannot be changed using the You-Are service, this test shall be skipped.~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	

Specified Test Changes

[For the following tests, move each from 135.1 into BTL Specified tests and change as noted. Note: items highlighted in green were changed by BTLWG-1389]

8.44 Who-Am-I Service Initiation Tests

8.44.1 Responds to Who-Is With Who-Am-I While in the Unconfigured State

Reason for change: Removed conditional skip.

Purpose: To verify the IUT will send a Who-Am-I while in the unconfigured state when a Who-Is is received.

Test Concept: TD sends a Who-Is using the wildcard instance of 4194303 and the IUT responds with a Who-Am-I Request.

Test Configuration: The IUT's Device object is not configured with an object instance number. The IUT is configured with a MAC address. ~~If the IUT does not support having a MAC address but no configured Device object instance number, this test shall be skipped.~~

Notes to Tester: The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. TRANSMIT
   DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   Who-Is Request,
   'Device Instance Range Low Limit' = 4194303,
   'Device Instance Range High Limit' = 4194303
2. BEFORE Unconfirmed Response Fail Time
   RECEIVE
   DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   Who-Am-I Request,
   'Vendor Identifier' = (the IUT's Vendor_Identifier),
   'Model Name' = (the IUT's Model_Name),
   'Serial Number' = (the IUT's Serial_Number)

© 2026 by BACnet International. All rights reserved.

8.45 You-Are Service Initiation Tests

8.45.1 Configures Other Device's MAC Address

Reason for change: DeviceIdentifier incorrectly included in the You-Are request. Removed MS/TP Subordinate support and fixed WAIT times.

Purpose: To verify the IUT can configure another device's MAC address using the You-Are service without relying on the Who-Am-I service.

Test Concept: The IUT configures TD with an appropriate MAC address without TD first sending a Who-Am-I.

Configuration Requirements: TD's Device object is configured with a known Device object instance number and no configured MAC address.

Notes to Tester: The IUT may require the tester to specify all the parameters needed to configure TD with You-Are, using the IUT's software. The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. MAKE (the IUT configure TD)
2. RECEIVE
   DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   You-Are Request,
   'Vendor Identifier' = (TD's Vendor_Identifier),
   'Model Name' = (TD's Model_Name),
   'Serial Number' = (TD's Serial_Number),
   ~~'Device Identifier' = (TD's Device object),~~
   'Device MAC Address' = (an appropriate MAC address)

~~3. IF (TD is not an MS/TP subordinate node)~~

3. BEFORE Unconfirmed Response Fail Time

4. TRANSMIT
   DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   I-Am Request,
   'Device Identifier' = (TD's Device object)
   'Max APDU Length Accepted' = (any valid value),
   'Segmentation Supported' = (any valid value),
   'Vendor Identifier' = (TD's Vendor_Identifier)

[Move tests from 135.1 to BTL Specified Tests and modify]

8.45.2 ~~8.X36.2~~ Configures Other Device's Object Instance Number and MAC Address

Reason for change: Removed MS/TP Subordinate support and fixed WAIT times.

Purpose: To verify the IUT can configure *a* ~~another~~ Device's object instance number and MAC address using the You-Are service without relying on the Who-Am-I service.

Test Concept: The IUT configures TD with an appropriate Device object instance number (X) and MAC address without TD first sending a Who-Am-I.

Configuration Requirements: TD is not configured with a Device object instance number and has no configured MAC address.

Notes to Tester: The IUT will require the tester to specify all the parameters needed to configure TD with You-Are, using the IUT's software. The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. MAKE (the IUT configure TD)

~~2. BEFORE Internal Processing Fail Time~~

2. RECEIVE

© 2026 by BACnet International. All rights reserved.

DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
You-Are Request,
'Vendor Identifier' = (TD's Vendor_Identifier),
'Model Name' = (TD's Model_Name),
'Serial Number' = (TD's Serial_Number),
'Device Identifier' = (Device, X),
'Device MAC Address' = (any valid MAC address)

~~3. IF (TD is not an MS/TP subordinate node)~~

3. BEFORE Unconfirmed Response Fail Time

4. TRANSMIT
DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
I-Am Request,
'Device Identifier' = (Device, X),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (TD's Vendor_Identifier)

9.43 Who-Am-I Service Execution Tests

9.43.1 Uses Who-Is to Configure Devices Supporting the Who-Am-I Service

Reason for change: Fixed fail times.

Purpose: To verify that the IUT can configure a device when it receives a Who-Am-I in response to a Who-Is that it sent.

Test Concept: The IUT sends a Who-Is and TD responds with a Who-Am-I. The IUT then configures the TD's Device object instance number (X) using the You-Are service.

Configuration Requirements: TD has a MAC address but is not configured with a Device object instance number.

Notes to Tester: The IUT may require the tester to specify a Device object instance number to assign TD, using the IUT's software. If the IUT requires this step to occur at the beginning of step 3, then the amount of time it takes the tester to enter this should not be counted towards the fail timer. The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. MAKE (the IUT transmit a Who-Is request to discover devices needing configuration)
2. RECEIVE
DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
Who-Is Request,
'Device Instance Range Low Limit' = 4194303,
'Device Instance Range High Limit' = 4194303
3. TRANSMIT
DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
Who-Am-I Request,
'Vendor Identifier' = (TD's Vendor_Identifier),
'Model Name' = (any valid Model Name),
'Serial Number' = (any valid Serial Number)

~~4. BEFORE Internal Processing Fail Time~~

4. BEFORE Unconfirmed Response Fail Time

5. RECEIVE
DESTINATION = TD ~~| GLOBAL BROADCAST | LOCAL BROADCAST | REMOTE BROADCAST~~,
You-Are Request,
'Vendor Identifier' = (TD's Vendor_Identifier),
'Model Name' = (the Model Name sent by TD in the previous step),
'Serial Number' = (the Serial Number sent by TD in the previous step),
'Device Identifier' = (Device, X),
'Device MAC Address' = (TD's MAC address, or absent)

~~6. IF (TD is not an MS/TP subordinate node)~~

6. BEFORE Unconfirmed Response Fail Time

© 2026 by BACnet International. All rights reserved.

7. TRANSMIT
   DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   I-Am Request,
   'Device Identifier' = (Device, X),
   'Max APDU Length Accepted' = (any valid value),
   'Segmentation Supported' = (any valid value),
   'Vendor Identifier' = (TD's Vendor_Identifier)

9.43.2 Configures Other Device's Object Instance Number

Reason for change: Fixed fail times.

Purpose: To verify the IUT can configure another Device's object instance number using the You-Are service.

Test Concept: TD sends a Who-Am-I, and the IUT configures it with an appropriate Device object instance number (X).

Configuration Requirements: TD is configured with a MAC address but no Device object instance number. The IUT is not actively discovering devices.

Notes to Tester: The IUT may require the tester to specify a Device object instance number to assign TD, using the IUT's software. If the IUT requires this step to occur between it receiving the Who-Am-I request and sending a You-Are request, then the amount of time it takes the tester to enter this should not be counted towards the fail timer. The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. TRANSMIT
   DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   Who-Am-I Request,
   'Vendor Identifier' = (TD's Vendor_Identifier),
   'Model Name' = (any valid Model Name),
   'Serial Number' = (any valid Serial Number)
~~2. BEFORE Internal Processing Fail Time~~
2. BEFORE Unconfirmed Response Fail Time
3. RECEIVE
   DESTINATION = TD~~ | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST~~,
   You-Are Request,
   'Vendor Identifier' = (TD's Vendor_Identifier),
   'Model Name' = (the Model Name sent by TD in the previous step),
   'Serial Number' = (the Serial Number sent by TD in the previous step),
   'Device Identifier' = (Device, X),
   'Device MAC Address' = (TD's MAC address, or absent)
~~4. IF (TD is not an MS/TP subordinate node)~~
4. BEFORE Unconfirmed Response Fail Time
5. TRANSMIT
   DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   I-Am Request,
   'Device Identifier' = (Device, X)
   'Max APDU Length Accepted' = (any valid value),
   'Segmentation Supported' = (any valid value),
   'Vendor Identifier' = (TD's Vendor_Identifier)

9.44 You-Are Service Execution Tests

9.44.1 Positive Tests

9.44.1.1 Configurable Device Object Instance Number

Reason for change: Removed MS/TP Subordinate support and removed test cannot be skipped.

© 2026 by BACnet International. All rights reserved.

Purpose: To verify the IUT will send a Who-Am-I while *the Device object instance number is unconfigured* ~~in the unconfigured state~~.

Test Concept: The IUT is made to send a Who-Am-I while unconfigured. TD configures the IUT's Device object instance number (X).

Configuration Requirements: The IUT is not configured with a Device object instance number. The IUT is configured with a MAC address. ~~If the IUT cannot be configured in this way, this test shall be skipped.~~

Notes to Tester: The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. MAKE (the IUT send a Who-Am-I)
2. RECEIVE
   DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   Who-Am-I Request,
   'Vendor Identifier' = (the IUT's Vendor_Identifier),
   'Model Name' = (the IUT's Model_Name),
   'Serial Number' = (the IUT's Serial_Number)
3. TRANSMIT
   DESTINATION = IUT ~~| LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST~~,
   You-Are Request,
   'Vendor Identifier' = (the IUT's Vendor_Identifier),
   'Model Name' = (the IUT's Model_Name),
   'Serial Number' = (the IUT's Serial_Number),
   'Device Identifier' = (Device, X),
   'Device MAC Address' = (IUT's MAC address, or absent)

~~4. IF (the IUT is not an MS/TP subordinate node)~~

4. BEFORE Unconfirmed Response Fail Time
5. RECEIVE
   DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   I-Am Request,
   'Device Identifier' = (Device, X),
   'Max APDU Length Accepted' = (any valid value),
   'Segmentation Supported' = (any valid value),
   'Vendor Identifier' = (the IUT's Vendor_Identifier)

9.44.1.2 Configurable MAC Address

Reason for change: Removed MS/TP Subordinate support, verify device instance is not changed, and fixed WAIT times.

Purpose: To verify the IUT can be configured with a MAC address using the You-Are service.

Test Concept: TD sends a You-Are request to configure TD's MAC address.

Configuration Requirements: The IUT is not configured with a MAC address *or MAC address (M0)*. The IUT is configured with a Device object instance number (X). ~~If the IUT cannot be configured in this way, this test shall be skipped.~~

Notes to Tester: The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. TRANSMIT
   DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   You-Are Request,
   'Vendor Identifier' = (the IUT's Vendor_Identifier),
   'Model Name' = (the IUT's Model_Name),
   'Serial Number' = (the IUT's Serial_Number),
   'Device Identifier' = (absent),

© 2026 by BACnet International. All rights reserved.

'Device MAC Address' = (*M1, any* ~~a~~ valid MAC address *not equal to M0*)
~~2. IF (the IUT is not an MS/TP subordinate node)~~
~~3. BEFORE Internal Processing Fail Time~~
2. BEFORE Unconfirmed Response Fail Time
3. RECEIVE
SOURCE = M1
DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
I-Am Request,
'Device Identifier' = (Device, X),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (the IUT's Vendor_Identifier)

9.44.1.3 Configurable Device Object Instance Number and MAC Address

Reason for change: Removed MS/TP Subordinate support, fixed WAIT times, and verify MAC Address changed.

Purpose: To verify the IUT can be configured with both a Device object instance number and MAC address using the You-Are service.

Test Concept: TD sends a You-Are request to configure the device with both a *new* Device object instance number (X*1*) and *new* a MAC address *(M1)*.

Configuration Requirements: The IUT is ~~not~~ configured with a Device object instance number *(X0) and* ~~or~~ a MAC address *(M0)*. ~~If the IUT cannot be configured in this way, this test shall be skipped.~~

Notes to Tester: The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. TRANSMIT
DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
You-Are Request,
'Vendor Identifier' = (the IUT's Vendor_Identifier),
'Model Name' = (the IUT's Model_Name),
'Serial Number' = (the IUT's Serial_Number),
'Device Identifier' = (Device, X*1, any valid value not equal to X0*),
'Device MAC Address' = (*M1,* any valid MAC address not equal to *M0*)
~~2. IF (the IUT is not an MS/TP subordinate node)~~
~~3. BEFORE Internal Processing Fail Time~~
2. BEFORE Unconfirmed Response Fail Time
3. RECEIVE
SOURCE = M1
DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
I-Am Request,
'Device Identifier' = (Device, X*1*),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (the IUT's Vendor_Identifier)

9.44.1.4 Device Object Instance Number is Configurable Even When MAC Address is Also Sent

Reason for change: Removed MS/TP Subordinate support and verify MAC Address unchanged.

Purpose: To verify the IUT can be configured with a Device object instance number even when a MAC address that does not match its own is also sent.

Test Concept: The ~~IUT sends a Who-Am-I while unconfigured.~~ TD assigns the IUT a Device object instance number (X) and also sends a MAC address (M1), which is different than the IUT's MAC address *(M0)*. The IUT will accept *the device identifier and may or may not save the new MAC address.* ~~both values, but not change its MAC address to M1.~~

© 2026 by BACnet International. All rights reserved.

Configuration Requirements: The IUT is not configured with a Device ID. The IUT is configured with a MAC address *(M0)*. ~~If the IUT cannot be configured in this way, this test shall be skipped. If the IUT cannot be configured with a MAC address that cannot be changed using You-Are, this test shall be skipped.~~

Notes to Tester: The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

~~1. MAKE (The IUT send a Who-Am-I)~~
~~2. RECEIVE~~
~~SOURCE = IUT~~
~~DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,~~
~~Who-Am-I Request,~~
~~'Vendor Identifier' = (the IUT's Vendor_Identifier),~~
~~'Model Name' = (the IUT's Model_Name),~~
~~'Serial Number' = (the IUT's Serial_Number)~~
1. TRANSMIT
DESTINATION = *M0*~~IUT~~ | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
You-Are Request,
'Vendor Identifier' = (the IUT's Vendor_Identifier),
'Model Name' = (the IUT's Model_Name),
'Serial Number' = (the IUT's Serial_Number),
'Device Identifier' = (Device, X),
'Device MAC Address' = (M1: a MAC address different *from M0*~~than that of the IUT~~)
~~2. IF (the IUT is not an MS/TP subordinate node)~~
2. BEFORE Unconfirmed Response Fail Time
3. RECEIVE
SOURCE = *M0 | (M1, if the IUT supports a changeable MAC Address)* ~~(if the IUT IUT ensure that the MAC address has not changed~~
DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
I-Am Request,
'Device Identifier' = (Device, X),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (the IUT's Vendor_Identifier)
~~5. CHECK (the IUT's MAC address did not change)~~

9.44.1.5 MAC Address is Reconfigurable

Reason for change: Fixed misc MAC Address and Device instance references.

Purpose: To verify the IUT can be configured with a MAC address different *from* ~~than~~ the one it was configured with using the You-Are service~~, without having to be unconfigured first~~.

Test Concept: The ~~IUT is configured and does not need a MAC address configured to it.~~ TD assigns a different MAC address to the IUT using the You-Are service. The IUT changes its MAC address.

Configuration Requirements: The IUT is configured with a MAC address *(M0)*. ~~If the IUT cannot be configured with a MAC address that can be changed using You-Are, this test shall be skipped.~~

Notes to Tester: The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. READ (X = Device,4194303, ObjectIdentifier)
~~1. VERIFY (Device, IUT), Object_Type = DEVICE~~
2. TRANSMIT
DESTINATION = *M0*~~IUT~~ | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
You-Are Request,

© 2026 by BACnet International. All rights reserved.

'Vendor Identifier' = (the IUT's Vendor_Identifier),
'Model Name' = (the IUT's Model_Name),
'Serial Number' = (the IUT's Serial_Number),
'Device Identifier' = (*X~~the IUT's Device Object_Identifier~~*),
'Device MAC Address' = (*M1,* a MAC address different than that of the IUT *(M0)*)

~~3. IF (the IUT is not an MS/TP subordinate node)~~
4. BEFORE Unconfirmed Response Fail Time
5. RECEIVE
SOURCE = *M1*~~IUT~~ -- ensure that the MAC address is what was set by TD
DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
I-Am Request,
'Device Identifier' = (*X~~the IUT's Device Object_Identifier~~*),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (the IUT's Vendor_Identifier)

~~6. TRANSMIT~~
~~DESTINATION = IUT -- using the newly configured MAC address~~
~~ReadProperty-Request,~~
~~'Object Identifier' = (Device, IUT),~~
~~'Property Identifier' = Object_Type~~
~~7. RECEIVE~~
~~SOURCE = IUT -- ensure that the MAC address is what was set by TD~~
~~ReadProperty-Ack,~~
~~'Object Identifier' = (Device, IUT),~~
~~'Property Identifier' = Object_Type,~~
~~'Property Value' = DEVICE~~

9.44.1.6 Retains Configuration Through Restarts

Reason for change: Fixed the BEFORE values and now checks the MAC Address. Eliminated unnecessary steps.

Purpose: To verify the IUT will retain a configured Device object instance number and MAC address*, if configurable,* across a restart.

Test Concept: The IUT is configured with a Device object instance number (X) and MAC address (M1) using You-Are. The IUT is then restarted and the Device object instance number and MAC address are verified to be the same that were configured at the beginning of the test.

Configuration Requirements: ~~The IUT is unconfigured.~~ *The IUT is not configured with a MAC address or contains MAC address (M0). The IUT is not configured with a Device object instance number.*

Notes to Tester: ~~If the IUT's MAC address cannot be changed using You-Are, use the IUT's MAC address in place of M1.~~ The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. TRANSMIT
DESTINATION = *M0*~~IUT~~ | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
You-Are Request,
'Vendor Identifier' = (the IUT's Vendor_Identifier),
'Model Name' = (the IUT's Model Name),
'Serial Number' = (the IUT's Serial Number),
'Device Identifier' = (Device, X),
'Device MAC Address' = (M1~~, or absent if the IUT does not support a configurable MAC address~~)

2. BEFORE Unconfirmed Response Fail Time
3. RECEIVE
SOURCE = M0 | (M1, if the IUT supports a changeable MAC Address)
DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
I-Am Request,
'Device Identifier' = (Device, X),

© 2026 by BACnet International. All rights reserved.

'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (the IUT's Vendor_Identifier)

4. MAKE (the IUT restart)
5. WAIT (for the IUT to restart)
6. TRANSMIT
DESTINATION = IUT|LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
Who-Is Request,
'Device Instance Range Low Limit' = 4194303,
'Device Instance Range High Limit' = 4194303
~~6. WAIT Internal Processing Fail Time~~
7. BEFORE Unconfirmed Response Fail Time
8. RECEIVE
SOURCE = M0 | (M1, if the IUT supports a changeable MAC Address)
DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
I-Am Request,
'Device Identifier' = (Device, X),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (the IUT's Vendor_Identifier)
9. CHECK (the IUT did not send a Who-Am-I Request)
~~10. TRANSMIT~~
~~DESTINATION = IUT| LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,~~
~~Who-Is Request,~~
~~12. RECEIVE~~
~~DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,~~
~~I-Am Request,~~
~~'Device Identifier' = (Device, X),~~
~~'Max APDU Length Accepted' = (any valid value),~~
~~'Segmentation Supported' = (any valid value),~~
~~'Vendor Identifier' = (the IUT's Vendor_Identifier)~~
~~13. CHECK (the IUT's MAC address is M1 if it was changeable)~~

9.44.1.7 Unconfigurable by You-Are

Reason for change: Fixed the WAIT values.

Purpose: To verify the IUT can be unconfigured with the You-Are service.

Test Concept: The IUT is configured with a Device object instance number and MAC address. TD then sends a You-Are request with 'Device Identifier' set to ~~4149303~~*4194303*, and the ~~IUT~~ *IUT's Device object instance number* becomes unconfigured.

Configuration Requirements: The IUT is configured with both a Device object instance number and MAC address *(M0)*.

Notes to Tester: The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. TRANSMIT
DESTINATION = ~~IUT~~*M0* | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
You-Are Request,
'Vendor Identifier' = (the IUT's Vendor_Identifier),
'Model Name' = (the IUT's Model Name),
'Serial Number' = (the IUT's Serial Number),
'Device Identifier' = (Device, 4194303)
'MAC Address' = (~~the IUT's MAC address~~*M0*, or absent)
2. WAIT Unconfirmed Response Fail Time
3. CHECK (the IUT did not send any requests, except for an optional Who-Am-I if the MAC address cannot be unconfigured)
4. TRANSMIT

© 2026 by BACnet International. All rights reserved.

DESTINATION = M0,
ReadProperty-Request,
'Object Identifier' = ~~(any object that would exist in the IUT, or~~ (Device, 4194303)~~),~~
'Property Identifier' = (any property in the object selected)
5. CHECK (the IUT did not respond with a ReadProperty-ACK)

9.44.2 Negative Tests

9.44.2.1 Only Supports Execution of the You-Are Service While Unconfigured

[Remove Test]
~~Reason for change: Fixed fail times.~~

~~Purpose: To verify the IUT does not initiate or execute any other services while the MAC address is unconfigured.~~

~~Test Concept: The IUT's MAC address is unconfigured and the tester attempts to~~ attempts to ~~execute any services it would otherwise execute.~~

~~Configuration Requirements:~~ The IUT is configured with a MAC address. ~~If the IUT's MAC address cannot be unconfigured, then this test shall be skipped.~~

~~Notes to Tester: The destination address used by TD shall be selected such that the IUT will receive the messages.~~

~~Test Steps:~~

~~1. TRANSMIT~~
~~DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,~~
~~You-Are-Request,~~
~~'Vendor Identifier' = (the IUT's Vendor_Identifier),~~
~~'Model Name' = (the IUT's Model_Name),~~
~~'Serial Number' = (the IUT's Serial_Number),~~
~~'Device Identifier' = (Device, 4194303)~~
~~'MAC Address' = (the IUT's MAC address, or absent)~~
~~2. WAIT Internal Processing Fail Time~~
~~*2. WAIT Unconfirmed Response Fail Time*~~
~~3. CHECK (the IUT did not transmit any requests)~~
~~4. TRANSMIT~~
~~DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,~~
~~ReadProperty-Request,~~
~~'Object Identifier' = (any object that would exist in the IUT, or (Device, 4194303)),~~
~~'Property Identifier' = (any property in the object selected)~~
~~5. WAIT Internal Processing Fail Time~~
~~*5. WAIT Acknowledgement Fail Time*~~
~~6. CHECK (the IUT did not respond with a ReadProperty-ACK)~~

9.44.2.2 Only Accepts Configuration When Received Parameters Match

Reason for change: Validates the device instance does not change.

Purpose: To verify the IUT will not ~~configure or~~ reconfigure itself when the parameters in a You-Are request do not match its vendor identifier, model name, and serial number.

Test Concept: The IUT is *configured* ~~unconfigured~~ and is sent a You-Are but with the wrong Vendor Identifier, Model Name, and Serial Number. The IUT does not accept the configuration and does not transmit an I-Am request indicating it has been configured.

Configuration Requirements: The IUT is configured with *Device object instance number and MAC address*. ~~needs configuration of either Device object instance number or MAC address, or both. This test shall be skipped if the IUT is an MS/TP subordinate node.~~

© 2026 by BACnet International. All rights reserved.

Notes to Tester: ~~If the IUT only supports configuration of either Device object instance number or MAC address but not both, TD shall use the IUT's Device Identifier or MAC address, whichever is configured, when sending You Are requests.~~ The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. READ (X = Device,4194303, ObjectIdentifier)
2. TRANSMIT
   DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   You-Are-Request,
   'Vendor Identifier' = (the IUT's Vendor_Identifier),
   'Model Name' = (the IUT's Model_Name),
   'Serial Number' = (any value other than the IUT's Serial_Number),
   'Device Identifier' = (any valid Device Identifier *not equal to X*),
   'Device MAC Address' = (~~any valid MAC address, or~~ absent)
3. WAIT Unconfirmed Response Fail Time
~~4. CHECK (the IUT did not transmit an I-Am-Request)~~
4. TRANSMIT
   DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   You-Are Request,
   'Vendor Identifier' = (the IUT's Vendor_Identifier),
   'Model Name' = (any value other than the IUT's Model_Name),
   'Serial Number' = (IUT's Serial_Number),
   'Device Identifier' = (any valid Device Identifier *not equal to X*),
   'Device MAC Address' = (~~any valid MAC address, or~~ absent)
5. WAIT Unconfirmed Response Fail Time
~~7. CHECK (the IUT did not transmit an I-Am-Request)~~
6. TRANSMIT
   DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   You-Are Request,
   'Vendor Identifier' = (any value other than the IUT's Vendor_Identifier),
   'Model Name' = (the IUT's Model_Name),
   'Serial Number' = (the IUT's Serial_Number),
   'Device Identifier' = (any valid Device Identifier *not equal to X*),
   'Device MAC Address' = (~~any valid MAC address, or~~ absent)
7. WAIT Unconfirmed Response Fail Time
~~10. CHECK (the IUT did not transmit an I-Am-Request)~~
8. VERIFY (Device, 4194303, ObjectIdentifier = X)

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp4-3: Update pTimeDelay and pTimeDelayNormal Tests [BTLWG-1826]

Overview:

Follow-up to BTLWG-1612, which resolved CR-0571. CR-0571 asked for clarification on how to proceed with test 135.1-2025 – 7.3.1.12, which includes a step to wait for pTimeDelay and pTimeDelayNormal; however, the test may be executed against algorithms that do not support pTimeDelay and pTimeDelayNormal. BTLWG-1612 resolved the issue in 135.1-2025 by adding an additional Notes to Tester.

This work item resolves the other impacted tests.

Changes:

Checklist Changes

None

Test Plan Changes

[Modify the Notification Class Object section Base Requirements]
[Modify the Notification Class Object section Supports Writable Recipient_List Property]
[Modify the Event Log Object section Base Requirements]
[Modify the Alarm and Event Management - Notification - Internal - B section Base Requirements]
[Modify the Alarm and Event Management - Notification - Internal - B section Supports Event_Message_Texts Property
[Modify the Alarm and Event Management - Notification - Internal - B section Supports Event_Message_Texts_Config Property]
[Modify the Alarm and Event Management - Acknowledge - B section Base Requirements]
[Modify the Alarm and Event Management - LifeSafety - B section Base Requirements]
[Modify the Alarm and Event Management - LifeSafety - B section Supports Event_Message_Texts Property]
[Modify the Alarm and Event Management - LifeSafety - B section Supports Event_Message_Texts_Config Property]
[Modify the Alarm and Event Management - Access. Control - B section Base Requirements]
[Modify the Alarm and Event Management - Access. Control - B section Supports Event_Message_Texts Property]
[Modify the Alarm and Event Management - Access. Control - B section Supports Event_Message_Texts_Config Property]
[Modify the Trending - Automated Trend Retrieval - B section Implements Intrinsic Alarming]
[Modify the Trending - Automated Multiple Value Retrieval - B section Implements Intrinsic Alarming]

3.17 Notification Class Object

[Move BTL - 7.3.2.21.3.1 ValidDays Test from Notification Class Object's Base Requirements Section to Notification Class Object's Supports Writable Recipient_List Property section; change ValidDays to Valid Days to match standard]
[Move 135.1-2025 - 7.3.2.21.3.2 - FromTime and ToTime Test from Notification Class Object's Base Requirements section to Notificaton Class Object's Supports Writable Recipient_List Property section; change FormTime to From Time and ToTime to To Time to match 135-2024 standard. Move to BTL Specified tests.]
[Modify the Notification Class Object's Supports Writable Recipient_List Property Section to move tests to BTL Specified Tests]
[Update fromTime and toTime in Supports Writable Recipient_List Property section of the Test Plan]

3.17.1 Base Requirements

Base requirements must be met by any IUT that can contain Notification Class objects.

~~**BTL - 7.3.2.21.3.1 - ValidDays Test**~~		
	~~**Test Conditionality**~~	~~Must be executed if the IUT keeps time.~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	
~~**135.1-2025 - 7.3.2.21.3.2 - FromTime and ToTime Test**~~		
	~~**Test Conditionality**~~	~~Must be executed if the IUT keeps time~~

© 2026 by BACnet International. All rights reserved.

	~~**Test Directives**~~	
	~~**Testing Hints**~~	
135.1-2025 - 7.3.2.21.3.4 - Transitions Test		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
…		

3.17.3 Supports Writable Recipient_List Property

The IUT supports Recipient_List properties that are modifiable via write services. All required values must be accepted in write requests.

Verify Checklist		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims support for DS-WP-B and 'Contains writable list properties' in the Checklist.
	Testing Hints	
135.1-2025 - 9.22.1.5 - Writing to Properties Based on Data Type		
	Test Conditionality	Must be executed.
	Test Directives	Verify that all required values of BACnetDestination can be written (into a list element).
	Testing Hints	
~~**135.1-2025**~~***BTL*** **- 7.3.2.21.3.5 - Recipient_List Property Supports Device Identifier Recipients Test**		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
~~**135.1-2025**~~***BTL*** **- 7.3.2.21.3.6 - Recipient_List Property Supports Network Address Recipients**		
	Test Conditionality	Must be executed.
	Test Directives	Execute test multiple times using unicast, local broadcast, remote broadcast and global broadcast addresses. Use unicast MAC addresses of 1-6 octets. Ensure that a recipient with the address form having a network number of 0 is not used when the IUT is configured as a BACnet router.
	Testing Hints	
135.1-2025 - 7.2.8 - Time Non-Pattern Properties Test		
	Test Conditionality	Must be executed if the IUT claims Protocol_Revision 11 or greater.
	Test Directives	Apply to the ~~fromTime~~*From Time* and again to the ~~toTime~~*To Time* element in a Recipient_List property in a Notification Class.
	Testing Hints	
135.1-2025 - 9.23.2.20 - Time Non-Pattern Properties Test using WritePropertyMultiple Service		
	Test Conditionality	This test shall only be applied to devices claiming Protocol_Revision 11 or higher and which supports execution of WritePropertyMultiple.
	Test Directives	Apply to the ~~fromTime~~*From Time* and again to the ~~toTime~~*To Time* element in a Recipient_List property in a Notification Class.
	Testing Hints	
BTL - 7.3.2.21.3.1 - Valid Days Test		
	Test Conditionality	*Must be executed if the Valid Days parameters support values other than all days*
	Test Directives	
	Testing Hints	
BTL - 7.3.2.21.3.2 - From Time and To Time Test		
	Test Conditionality	*Must be executed if the From Time and To Time parameters support values other than 0:00:00.0 and 23:59:59.99 respectively.*
	Test Directives	
	Testing Hints	

© 2026 by BACnet International. All rights reserved.

3.22 Event Log Object

[Modify the Event Log Object section Base Requirements]

3.22.4 Supports Buffer Ready Notification

The IUT supports the BUFFER_READY notification from the Event Log object.

…		
135.1-2025 - 8.5.7 - BUFFER_READY Tests (UnconfirmedEventNotification)		
	Test Conditionality	If BUFFER_READY Notification is tested through **8.4.7,** then this test can be skipped.
	Test Directives	The 'Event Object Identifier' in this test must be either an Event Log object contained in the IUT or an Event Enrollment object in the IUT monitoring the Event Log Object contained in the IUT.
	Testing Hints	
~~135.1-2025~~*BTL* - 7.3.1.10.2 - Event_Enable Tests for TO_NORMAL only Algorithms		
	Test Conditionality	If the IUT cannot be configured to meet the configuration requirements then this test shall be skipped.
	Test Directives	The 'Event Object Identifier' in this test must be either an Event Log object contained in the IUT or an Event Enrollment object in the IUT monitoring the Event Log Object contained in the IUT.
	Testing Hints	
135.1-2025 - 7.3.1.12 - Notify_Type Test		
	Test Conditionality	If the IUT cannot be configured to meet the configuration requirements then this test shall be skipped.
	Test Directives	The 'Event Object Identifier' in this test must be either an Event Log object contained in the IUT or an Event Enrollment object in the IUT monitoring the Event Log Object contained in the IUT.
	Testing Hints	
…		

5.2 Alarm and Event Management - Notification - Internal - B

[Modify the Alarm and Event Management - Notification - Internal - B section Base Requirements]
[Modify the Alarm and Event Management - Notification - Internal - B section Supports Event_Message_Texts Property
[Modify the Alarm and Event Management - Notification - Internal - B section Supports Event_Message_Texts_Config Property]

5.2.1 Base Requirements

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

…		
135.1-2025 - 7.3.1.22.2 - Event_Detection_Enable Inhibits FAULT		
	Test Conditionality	If Protocol_Revision < 13 or if the IUT doesn't contain any event generating objects which support fault detection, and can be configured with an Event_Detection_Enable value of FALSE, then this test shall be skipped.
	Test Directives	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.
	Testing Hints	
~~135.1-2025~~*BTL* - 7.3.1.19.1 - Event_Algorithm_Inhibit Test		
	Test Conditionality	Apply this test when the Event_Algorithm_Inhibit property is present in an object which does not support the Event_Algorithm_Inhibit_Ref property or where the Event_Algorithm_Inhibit_Ref property is not

© 2026 by BACnet International. All rights reserved.

		initialized. If the IUT cannot be configured to contain such an object, then this test shall be skipped.
	Test Directives	The object types 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.
	Testing Hints	
135.1-2025 - 7.3.1.20.1 - Event_Algorithm_Inhibit_Ref Test		
	Test Conditionality	If the IUT has no object in which the Event_Algorithm_Inhibit_Ref property is present this test shall be skipped.
	Test Directives	The object types 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.
	Testing Hints	
…		

5.2.27 Supports Event_Message_Texts Property

The IUT contains one or more objects that support the Event_Message_Texts property.

~~135.1-2025~~*BTL* **- 7.3.1.17 - Event_Message_Texts Tests**		
	Test Conditionality	Must be executed.
	Test Directives	Repeat test once for each object type in the IUT that contains an Event_Message_Texts property.
	Testing Hints	

5.2.28 Supports Event_Message_Texts_Config Property

The IUT contains one or more objects that support the Event_Message_Texts_Config property.

~~135.1-2025~~*BTL* **- 7.3.1.18 - Event_Message_Texts_Config Test**		
	Test Conditionality	Must be executed.
	Test Directives	Repeat for each supported transition type (TO_OFFNORMAL, TO_FAULT, TO_NORMAL). Different objects may be selected for different transitions.
	Testing Hints	

5.5 Alarm and Event Management - Acknowledge - B

[Modify the Alarm and Event Management - Acknowledge - B section Base Requirements]

5.5.1 Base Requirements

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

…		
135.1-2025 - 9.1.1.11 - Successful Alarm Re-Acknowledgment of Unconfirmed Event Notifications		
	Test Conditionality	If the IUT supports revision 7 or higher this test must be executed.
	Test Directives	
	Testing Hints	
~~135.1-2025~~*BTL* **- 7.3.1.11.1 - Acked_Transitions Test**		
	Test Conditionality	Must be executed if the IUT supports normal to offnormal event generating objects.
	Test Directives	
	Testing Hints	
BTL - 7.3.1.11.4 - Acked_Transitions Test for Normal to Fault Transitions		

© 2026 by BACnet International. All rights reserved.

	Test Conditionality	Must be executed if the IUT supports normal to fault event generating objects.
	Test Directives	
	Testing Hints	
135.1-2025 - 7.3.1.19.3 - Event_Algorithm_Inhibit Acknowledgement Test		
	Test Conditionality	If the IUT has no object in which the Event_Algorithm_Inhibit property is present or has no object in which Event_Detection_Enable can be made TRUE, this test shall be skipped. If the IUT cannot be configured to contain any object with an unacknowledged event, then this test shall be skipped.
	Test Directives	The object types 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.
	Testing Hints	
135.1-2025 - 9.1.1.15 - Unsupported Message Text Character Set Acknowledge Alarm Test		
	Test Conditionality	If the IUT supports all character sets, this test shall be skipped.
	Test Directives	Repeat for every unsupported character set. Use at least one ConfirmedEventNotification and one UnconfirmedEventNotification.
	Testing Hints	
~~135.1-2025~~*BTL* - 7.3.1.11.3 - Acked_Transitions Test for Normal-To-Normal Transitions		
	Test Conditionality	Must be executed if the IUT supports normal to normal event generating objects.
	Test Directives	
	Testing Hints	

5.19 Alarm and Event Management - LifeSafety - B

[Modify the Alarm and Event Management - LifeSafety - B section Base Requirements]
[Modify the Alarm and Event Management - LifeSafety - B section Supports Event_Message_Texts Property]
[Modify the Alarm and Event Management - LifeSafety - B section Supports Event_Message_Texts_Config Property]

5.19.1 Base Requirements

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

…		
135.1-2025 - 7.3.1.22.2 - Event_Detection_Enable Inhibits FAULT		
	Test Conditionality	If Protocol_Revision < 13 or if the IUT doesn't contain any event generating objects which support fault detection, then this test shall be skipped.
	Test Directives	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.
	Testing Hints	
~~135.1-2025~~*BTL* - 7.3.1.19.1 - Event_Algorithm_Inhibit Test		
	Test Conditionality	If the IUT has no object which generates CHANGE_OF_LIFE_SAFETY notifications in which the Event_Algorithm_Inhibit property is present and does not support the Event_Algorithm_Inhibit_Ref property or where the Event_Algorithm_Inhibit_Ref property is not initialized, then this test shall be skipped.
	Test Directives	The object types 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.
	Testing Hints	

© 2026 by BACnet International. All rights reserved.

135.1-2025 - 7.3.1.20.1 - Event_Algorithm_Inhibit_Ref Test		
	Test Conditionality	If the IUT has no object in which the Event_Algorithm_Inhibit_Ref property is present or has no object in which Event_Detection_Enable can be made TRUE, this test shall be skipped.
	Test Directives	The object types 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.
	Testing Hints	
…		

5.19.11 Supports Event_Message_Texts Property

The IUT contains one or more objects that support the Event_Message_Texts property.

~~135.1-2025~~*BTL* - 7.3.1.17 - Event_Message_Texts Tests		
	Test Conditionality	Must be executed.
	Test Directives	Repeat test once for each object type in the IUT that contains an Event_Message_Texts property.
	Testing Hints	

5.19.12 Supports Event_Message_Texts_Config Property

The IUT contains one or more objects that support the Event_Message_Texts_Config property.

~~135.1-2025~~*BTL* - 7.3.1.18 - Event_Message_Texts_Config Test		
	Test Conditionality	Must be executed.
	Test Directives	Repeat for each supported transition type (TO_OFFNORMAL, TO_FAULT, TO_NORMAL). Different objects may be selected for different transitions.
	Testing Hints	

5.29 Alarm and Event Management - Access Control - B

[Modify the Alarm and Event Management - Access. Control - B section Base Requirements]
[Modify the Alarm and Event Management - Access. Control - B section Supports Event_Message_Texts Property]
[Modify the Alarm and Event Management - Access. Control - B section Supports Event_Message_Texts_Config Property]

5.29.1 Base Requirements

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

Verify Checklist		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT supports Access Point objects.
	Testing Hints	
~~135.1-2025~~*BTL* - 7.3.1.10.2 - Event_Enable Tests for TO_NORMAL only Algorithms		
	Test Conditionality	Must be executed.
	Test Directives	Apply to an object which generates ACCESS_EVENT notifications.
	Testing Hints	
135.1-2025 - 7.3.1.12 - Notify_Type Test		
	Test Conditionality	If the IUT cannot be configured to meet the configuration requirements then this test shall be skipped.
	Test Directives	Apply to an object which generates ACCESS_EVENT notifications.
	Testing Hints	
BTL - 7.3.1.22.1 - Event_Detection_Enable Inhibits Event Generation		

© 2026 by BACnet International. All rights reserved.

	Test Conditionality	If Protocol_Revision < 13, then this test shall be skipped. If Event_Detection_Enable cannot be set to FALSE, this test shall be skipped.
	Test Directives	Apply to an object which generates ACCESS_EVENT notifications.
	Testing Hints	
135.1-2025 - 7.3.1.22.2 - Event_Detection_Enable Inhibits FAULT		
	Test Conditionality	If Protocol_Revision < 13, or if the IUT doesn't contain any ACCESS_EVENT objects which support fault detection and can be configured with an Event_Detection_Enable value of FALSE, then this test shall be skipped.
	Test Directives	Apply to an object which generates ACCESS_EVENT notifications.
	Testing Hints	
~~135.1-2025~~*BTL* - 7.3.1.19.1 - Event_Algorithm_Inhibit Test		
	Test Conditionality	If the IUT has no object which generates ACCESS_EVENT notifications in which the Event_Algorithm_Inhibit property is present and does not support the Event_Algorithm_Inhibit_Ref property or where the Event_Algorithm_Inhibit_Ref property is not initialized, then ~~this test shall be skipped.~~.*this test shall be skipped.*
	Test Directives	Apply to an object which generates ACCESS_EVENT notifications.
	Testing Hints	
135.1-2025 - 7.3.1.20.1 - Event_Algorithm_Inhibit_Ref Test		
	Test Conditionality	If the IUT has no object which generates ACCESS_EVENT notifications in which the Event_Algorithm_Inhibit_Ref property is present, this test shall be skipped.
	Test Directives	
	Testing Hints	
…		

5.29.10 Supports Event_Message_Texts Property

The IUT supports Access Point objects that support the Event_Message_Texts property.

~~135.1-2025~~*BTL* - 7.3.1.17 - Event_Message_Texts Tests		
	Test Conditionality	Must be executed.
	Test Directives	Apply to an object which generates ACCESS_EVENT notifications that contains an Event_Message_Texts property.
	Testing Hints	

5.29.11 Supports Event_Message_Texts_Config Property

The IUT supports Access Point objects that support the Event_Message_Texts_Config property.

135.1-2025*BTL* - 7.3.1.18 - Event_Message_Texts_Config Test		
	Test Conditionality	Must be executed.
	Test Directives	Apply to an object which generates ACCESS_EVENT notifications. Repeat for each supported transition type (TO_OFFNORMAL, TO_FAULT, TO_NORMAL). Different objects may be selected for different transitions.
	Testing Hints	

7.6 Trending - Automated Trend Retrieval - B

[Modify the Trending - Automated Trend Retrieval - B section Implements Intrinsic Alarming]

7.6.4 Implements Intrinsic Alarming

The IUT contains, or can be made to contain, a Trend Log object that can generate BUFFER_READY notifications.

© 2026 by BACnet International. All rights reserved.

Each Trend Log object must include all of the Trend Log properties required for intrinsic reporting, i.e. Notification_Threshold, Records_Since_Notification, Last_Notify_Record, Notification_Class, Event_Enable, Acked_Transitions, Notify_Type, & Event_Time_Stamps.

~~135.1-2025~~*BTL* - 7.3.1.10.2 - Event_Enable Tests for TO_NORMAL only Algorithms		
	Test Conditionality	If the IUT cannot be configured to meet the configuration requirements then this test shall be skipped.
	Test Directives	
	Testing Hints	
135.1-2025 - 7.3.1.12 - Notify_Type Test		
	Test Conditionality	If the IUT cannot be configured to meet the nfiguration requirements then this test shall be skipped.
	Test Directives	
	Testing Hints	
…		

7.10 Trending - Automated Multiple Value Retrieval - B

[Modify the Trending - Automated Multiple Value Retrieval - B section Implements Intrinsic Alarming]

7.10.4 Implements Intrinsic Alarming

The IUT contains, or can be made to contain, a Trend Log Multiple object that can generate BUFFER_READY notifications.

Each Trend Log Multiple object must include all of the properties required for intrinsic reporting, i.e. Notification_Threshold, Records_Since_Notification, Last_Notify_Record, Notification_Class, Event_Enable, Acked_Transitions, Notify_Type, & Event_Time_Stamps.

~~135.1-2025~~*BTL* - 7.3.1.10.2 - Event_Enable Tests for TO_NORMAL only Algorithms		
	Test Conditionality	If the IUT cannot be configured to meet the configuration requirements then this test shall be skipped.
	Test Directives	
	Testing Hints	
135.1-2025 - 7.3.1.12 - Notify_Type Test		
	Test Conditionality	If the IUT cannot be configured to meet the configuration requirements then this test shall be skipped.
	Test Directives	
	Testing Hints	
…		

Specified Test Changes

[Move test 7.3.1.10.2 from 135.1-2025 into BTL Specified Tests, and modify]
[Move test 7.3.1.11.1 from 135.1-2025 into BTL Specified Tests, and modify]
[Move test 7.3.1.11.3 from 135.1-2025 into BTL Specified Tests, and modify]
[Move test 7.3.1.17 from 135.1-2025 into BTL Specified Tests, and modify]
[Move test 7.3.1.18 from 135.1-2025 into BTL Specified Tests, and modify]
[Move test 7.3.1.19 from 135.1-2025 into BTL Specified Tests, and modify]
[Move test 7.3.2.21.3.2 from 135.1-2025 into BTL Specified Tests, and modify]
[Move test 7.3.2.21.3.4 from 135.1-2025 into BTL Specified Tests, and modify]
[Move test 7.3.2.21.3.5 from 135.1-2025 into BTL Specified Tests, and modify]
[Move test 7.3.2.21.3.6 from 135.1-2025 into BTL Specified Tests, and modify]

7.3.1.10.2 Event_Enable Tests for TO_NORMAL only Algorithms

© 2026 by BACnet International. All rights reserved.

Reason For Change: Add new Notes to Tester to include a note on how to execute the test with algorithms that do not support the pTimeDelayNormal parameter. Move existing Notes to Tester to exist above Test Steps. Renumber to use new number schema and fix typo in Step 7.

Purpose: To verify that notification messages are transmitted only if the bit in Event_Enable corresponding to the event transition has a value of TRUE. This test applies to objects that only support generation of TO_NORMAL transitions.

Test Concept: The IUT is configured such that the Event_Enable property indicates that some event transitions are to trigger an event notification. Each event transition is triggered, and the IUT is monitored to verify that notification messages are transmitted only for those transitions for which the Event_Enable property has a value of TRUE.

Configuration Requirements: In the Notification Class object providing recipient information, the value of the Transitions parameter for all recipients shall be (TRUE, TRUE, TRUE). The event-generating object shall be in a NORMAL state at the start of the test. D1 is either the pTimeDelayNormal parameter or, in case of the EXTENDED and proprietary algorithms, a vendor specific delay (may be zero).

Notes to Tester: For algorithms that do not apply the pTimeDelayNormal parameter, assume D1 equals zero when executing the test.

Notes to Tester: The UnconfirmedEventNotification service may be substituted for the ConfirmedEventNotification service, in which case the tester shall skip all of the steps in which a BACnet-SimpleACK-PDU is sent.

Test Steps:

1. VERIFY pCurrentState = NORMAL
2. MAKE (the TO-NORMAL bit of the Event_Enable property equal to TRUE)
3. MAKE (a condition exist that will cause the object to generate a TO_NORMAL transition)
4. WAIT D1
5. BEFORE **Notification Fail Time**

6. RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the event-generating object configured for this test),
'Time Stamp' = (any valid time stamp),
'Notification Class' = (the class corresponding to the object being tested),
'Priority' = (the value configured to correspond to a TO-NORMAL transition),
'Event Type' = (any valid event type),
'Message Text' = (optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = NORMAL,
'Event Values' = (values appropriate to the event type)

~~*6.*~~*7.* ~~TRANSMITBACnet-SimpleACK-PDU~~*TRANSMIT BACnet-SimpleACK-PDU*

~~*7.*~~*8.* VERIFY pCurrentState = NORMAL

~~*8.*~~*9.* IF (Event_Enable can be changed such that the TO-NORMAL transition is FALSE) THEN {

10. MAKE (the TO-NORMAL bit of the Event_Enable property equal to FALSE)

11. MAKE (a condition exist that would cause the object to generate a TO_NORMAL transition)

12. WAIT (D1 + **Notification Fail Time**)

13. CHECK (verify that the IUT did not transmit an event notification message)

14. MAKE (the TO_NORMAL bit of the Event_Enable property equal to TRUE)

}

~~*9.*~~*15.* IF (the event-generating object can be placed into a fault condition) THEN {

16. IF (Event_Enable can be modified) THEN

17. MAKE (Event_Enable TO-FAULT transition equal TRUE)

18. IF (Event_Enable TO-FAULT transition = TRUE) THEN {

19. MAKE (the event-triggering object change to a fault condition)

20. BEFORE **Notification Fail Time**

21. RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),

© 2026 by BACnet International. All rights reserved.

'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the event-generating object configured for this test),
'Time Stamp' = (any valid time stamp),
'Notification Class' = (the class corresponding to the object being tested),
'Priority' = (the value configured to correspond to a TO-FAULT transition),
'Event Type' = (IF (Protocol_Revision < 13) THEN
(any valid event type),
ELSE
CHANGE_OF_RELIABILITY),
'Message Text' = (optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = FAULT,
'Event Values' = (values appropriate to the event type)

22. TRANSMIT BACnet-SimpleACK-PDU
23. VERIFY pCurrentState = FAULT
24. MAKE (the event-generating object change to a normal condition)
25. BEFORE **Notification Fail Time**
26. RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the event-generating object configured for this test),
'Time Stamp' = (any valid time stamp),
'Notification Class' = (the class corresponding to the object being tested),
'Priority' = (the value configured to correspond to a TO-NORMAL transition),
'Event Type' = (IF (Protocol_Revision < 13) THEN
(any valid event type),
ELSE
CHANGE_OF_RELIABILITY),
'Message Text' = (optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = FAULT,
'To State' = NORMAL,
'Event Values' = (values appropriate to the event type)

27. TRANSMIT BACnet-SimpleACK-PDU
28. VERIFY pCurrentState = NORMAL
}
}
~~10.~~*29.* IF (Event_Enable can be modified) THEN
30. MAKE (Event_Enable TO-FAULT transition equal FALSE)
~~11.~~*31.* IF (Event_Enable TO-FAULT transition = FALSE) THEN {
32. MAKE (the event-triggering object change to a fault condition)
33. VERIFY pCurrentState = FAULT
34. CHECK (verify that the IUT did not transmit an event notification message)
35. MAKE (the event-triggering object change to a normal condition)
}

~~Notes to Tester: The UnconfirmedEventNotification service may be substituted for the ConfirmedEventNotification service, in which case the TD shall skip all of the steps in which a BACnet-SimpleACK-PDU is sent.~~

7.3.1.11.1 Acked_Transitions Test

Reason For Change: Add new Notes to Tester to include a note on how to execute the test with the EXTENDED algorithm, which does not support the pTimeDelay or pTimeDelayNormal parameters. Add Test Conditionality for skipping test if the IUT only supports normal to normal event generating objects. Fix small typo in existing Notes to Tester.

Purpose: To verify that the Acked_Transitions property tracks whether or not an acknowledgment has been received for a previously issued event notification. It also verifies the interrelationship between Status_Flags and Event_State.

© 2026 by BACnet International. All rights reserved.

Test Concept: The IUT is configured such that the Event_Enable property indicates that all event transitions are to trigger an event notification. The Acked_Transitions property shall have the value (TRUE, TRUE, TRUE) indicating that all previous transitions have been acknowledged. Each 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 if an acknowledgment is received.

Test Conditionality: If the IUT only supports normal to normal event generating objects, then this test shall be skipped.

Configuration Requirements: The Event_Enable and Acked_Transitions properties shall be configured with a value of (TRUE, TRUE, TRUE). For analog objects the Limit_Enable property shall be configured with the value (TRUE, 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, TRUE, TRUE).

Notes to Tester: In case of the EXTENDED and proprietary algorithms, apply the vendor specific delay, which may equal zero, for the pTimeDelay and pTimeDelayNormal parameters when executing the test.

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 notifications.

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, TRUE, TRUE)
3. IF (Protocol_Revision is present AND Protocol_Revision >= 13) THEN
4. VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
5. IF (pMonitoredValue is writable) THEN
6. WRITE pMonitoredValue = (a value that is OFFNORMAL)
   ELSE
7. MAKE (pMonitoredValue have a value that is OFFNORMAL)
8. WAIT (pTimeDelay)
9. BEFORE **Notification Fail Time**
10. RECEIVE ConfirmedEventNotification-Request,
    'Process Identifier' = (~~P11: any valid process ID~~*any valid process ID*),
    'Initiating Device Identifier' = IUT,
    'Event Object Identifier' = (the event-generating object configured for this test),
    'Time Stamp' = (Toffnormal: any valid time stamp),
    'Notification Class' = (the class corresponding to the object being tested),
    'Priority' = (Poffnormal: the value configured to correspond to a TO-OFFNORMAL transition),
    'Event Type' = (any valid event type),
    'Message Text' = (optional, any valid message text),
    'Notify Type' = (the notify type configured for this event),
    'AckRequired' = TRUE,
    'From State' = NORMAL,
    'To State' = OFFNORMAL,
    'Event Values' = (values appropriate to the event type)
11. TRANSMIT BACnet-SimpleACK-PDU
12. VERIFY pCurrentState = OFFNORMAL
13. VERIFY Acked_Transitions = (FALSE, TRUE, TRUE)
14. IF (Protocol_revision is present AND Protocol_Revision >= 13 THEN
15. VERIFY Status_Flags = (TRUE, FALSE, ?, ?)
16. IF (pMonitoredValue is writable) THEN
17. WRITE pMonitoredValue = (a value that is NORMAL)
    ELSE
18. MAKE (pMonitoredValue have a value that is NORMAL)
19. WAIT (pTimeDelayNormal)
20. BEFORE **Notification Fail Time**
21. RECEIVE ConfirmedEventNotification-Request,

© 2026 by BACnet International. All rights reserved.

'Process Identifier' = (~~PI2: any valid process ID~~*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' = OFNORMAL,
'To State' = NORMAL,
'Event Values' = (values appropriate to the event type)

22. TRANSMIT BACnet-SimpleACK-PDU
23. VERIFY pCurrentState = NORMAL
24. VERIFY Acked_Transitions = (FALSE, TRUE, FALSE)
25. IF (Protocol_Revision is present AND Protocol_Revision >= 13) THEN
26. VERIFY Status_Flags = (FALSE, FALSE, ?,?)
27. TRANSMIT AcknowledgeAlarm-Request,
'Acknowledging Process Identifier' = (~~PI2~~*any valid process ID*),
'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)
28. RECEIVE BACnet-SimpleACK-PDU
29. IF (Protocol_Revision is present and Protocol_Revision >= 1) THEN
30. **BEFORE Notification Fail Time**
31. 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' = (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

32. BEFORE **Notification Fail Time**
33. 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' = (the IUT's current time or sequence number),
'Notification Class' = (the class corresponding to the object bind tested),
'Priority' = (Pnormal),
'Event Type' = (any valid event type),
'Message Text' = (optional, any valid message text),
'Notify Type' = ACK_NOTIFICATION
34. TRANSMIT BACnet-SimpleACK-PDU
35. VERIFY Acked_Transitions = (FALSE, TRUE, TRUE)
36. TRANSMIT AcknowledgeAlarm-Request,
'Acknowledging Process Identifier' = (~~PI1~~*any valid process ID*),
'Event Object Identifier' = (the event-generating object configured for this test),
'Event State Acknowledged' = OFFNORMAL,
'Time Stamp' = (Toffnormal),
'Acknowledgement Source' = (a character string),
'Time of Acknowledgment' = (the TD's current time)

© 2026 by BACnet International. All rights reserved.

37. RECEIVE BACnet-SimpleACK-PDU
38. IF (Protocol_Revision is present and Protocol_Revision >= 1) THEN
39. BEFORE **Notification Fail Time**
40. RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (~~PI~~*any valid process ID*),
'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' = (Poffnormal),
'Event Type' = (any valid event type),
'Message Text' = (optional, any valid message text),
'Notify Type' = ACK_NOTIFICATION,
'To State' = OFFNORMAL

ELSE

41. BEFORE **Notification Fail Time**
42. RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (~~PI~~*any valid process ID*),
'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' = (Poffnormal),
'Event Type' = (any valid event type),
'Message Text' = (optional, any valid message text),
'Notify Type' = ACK_NOTIFICATION
43. TRANSMIT BACnet-SimpleACK-PDU
44. VERIFY Acked_Transitions = (TRUE, TRUE, TRUE)

7.3.1.11.3 Acked_Transitions Test for Normal-To-Normal Transitions

Reason For Change: Add new Notes to Tester to indicate how to execute the tests with algorithms that do not support the pTimeDelay or pTimeDelayNormal parameters. Fix numbering issue.

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: For algorithms that do not apply the pTimeDelay or pTimeDelayNormal parameters, assume pTimeDelay or pTimeDelayNormal equals zero when executing the test.

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

© 2026 by BACnet International. All rights reserved.

4. VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
~~4.~~*5.* IF (pMonitoredValue is writable) THEN
~~5.~~*6.* WRITE pMonitoredValue = (a value that will result in a TO_NORMAL transition)
ELSE
~~6.~~*7.* MAKE (pMonitoredValue a value that will result in a TO_NORMAL transition)
~~7.~~*8.* WAIT (pTimeDelayNormal)
~~8.~~*9.* BEFORE **Notification Fail Time**
~~9.~~*10.* RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (~~PI2: any valid process ID~~*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)
~~10.~~*11.* TRANSMIT BACnet-SimpleACK-PDU
~~11.~~*12.* VERIFY pCurrentState = NORMAL
~~12.~~*13.* VERIFY Acked_Transitions = (?, ?, FALSE)
~~13.~~*14.* IF (Protocol_Revision is present AND Protocol_Revision >= 13) THEN
~~14.~~*15.* VERIFY Status_Flags = (FALSE, FALSE, ?,?)
~~15.~~*16.* 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)
~~16.~~*17.* RECEIVE BACnet-SimpleACK-PDU
~~17.~~*18.* IF (Protocol_Revision is present AND Protocol_Revision >= 1) THEN
~~18.~~*19.* BEFORE **Notification Fail Time**
~~19.~~*20.* 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' = (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
~~20.~~*21.* BEFORE **Notification Fail Time**
~~21.~~ *22.* 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' = (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
~~22.~~*23.* TRANSMIT BACnet-SimpleACK-PDU

© 2026 by BACnet International. All rights reserved.

~~23.~~*24.* VERIFY Acked_Transitions = (?, ?, TRUE)

7.3.1.17 Event_Message_Texts Tests

Reason For Change: Add new Notes to Tester to indicate how to execute the tests with algorithms that do not support the pTimeDelay or pTimeDelayNormal parameters. Fix spelling errors. Update test to match formatting of other tests, including updating to the new numbering schema.

Purpose: To verify that the value of the Event_Message_Texts property is updated when an object generates an event notification.

Test Concept: Read the Event_Message_Texts from the object. Transition the object through each event state which is enabled in the object saving the Message Text parameter from the received notification. Verify that the Event_Message_Texts updates with the Event_Message_Texts value received from the notification.

Configuration Requirements: The IUT shall be configured with an event-generation object, O1 which shall be in a NORMAL Event_State at the beginning of the test, and if Event_Enable is configurable, it shall have all bits set to TRUE for which the object supports ~~transtions.~~*transitions.*

Notes to Tester: For algorithms that do not apply the pTimeDelay or pTimeDelayNormal parameters, assume pTimeDelay or pTimeDelayNormal equals zero when executing the test.

Test Steps:
1. READ EMT = Event_Message_Texts
2. IF (Event_Enable is (TRUE, ?, ?) ~~and~~*AND* O1 can generate TO_OFFNORMAL transitions) THEN {
3. IF (pMonitoredValue is writable) THEN
4. WRITE pMonitoredValue = (a value that is offnormal)
ELSE
5. MAKE (pMonitoredValue a value that is offnormal)
6. WAIT (pTimeDelay)
7. BEFORE **Notification Fail Time**
8. RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = O1,
'Time Stamp' = (any valid timestamp),
'Notification Class' = (the class corresponding to the object being tested),
'Priority' = (the configured TO_OFFNORMAL priority),
'Event Type' = (any valid event type),
'Message Text' = (M: any valid value placed into EMT[1]),
'Notify Type' = Notify_Type,
'AckRequired' = (the configured value for the TO_OFFNORMAL transition),
'From State' = NORMAL,
'To State' = (any valid offnormal state),
'Event Values' = (values appropriate to the event type)
9. VERIFY Event_Message_Texts = EMT
}
~~3.~~*11.* IF (Event_Enable is (?, ?, TRUE) ~~and~~*AND* O1 can generate TO_NORMAL transitions) THEN {
12. IF (pMonitoredValue is writable) THEN
13. WRITE pMonitoredValue = (a value that will result in a TO_NORMAL transition)
ELSE
14. MAKE (pMonitoredValue a value that will result in a TO_NORMAL transition)
15. WAIT (pTimeDelayNormal)
16. BEFORE **Notification Fail Time**
17. RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = O1,
'Time Stamp' = (any valid time stamp),
'Notification Class' = (the class corresponding to the object being tested),

© 2026 by BACnet International. All rights reserved.

'Priority' = (the configured TO_NORMAL priority),
'Event Type' = (any valid event type),
'Message Text' = (M: any valid value placed into EMT[3]),
'Notify Type' = Notify_Type,
'AckRequired' = (the configured value for the TO_NORMAL transition),
'From State' = (any valid value),
'To State' = NORMAL,
'Event Values' = (values appropriate to the event type)

18. VERIFY Event_Message_Texts = EMT

}

~~4.~~*19.* IF (Event_Enable is (?, TRUE, ?) and O1 can generate TO_FAULT transitions) THEN {

20. MAKE (a condition exist that will cause O1 to generate a TO_FAULT transition)

21. BEFORE **Notification Fail Time**

22. RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = O1,
'Time Stamp' = (any valid time stamp),
'Notification Class' = (the class corresponding to the object being tested),
'Priority' = (the configured TO_FAULT priority),
'Event Type' = (IF (Protocol_Revision < 13 THEN
(any valid event type),
ELSE
(CHANGE_OF_RELIABILITY,
'Message Text' = (M: any valid value placed into EMT[2]),
'Notify Type' = Notify_Type,
'AckRequired' = (the configured value for the TO_FAULT transition),
'From State' = (any valid value),
'To State' = FAULT,
'Event Values' = (values appropriate to the event type)

23. VERIFY Event_Message_Texts = EMT

}

7.3.1.18 Event_Message_Texts_Config Test

Reason For Change: Add new Notes to Tester to indicate how to execute the tests with algorithms that do not support the pTimeDelay or pTimeDelayNormal parameters. Move Notes to Tester at end of test to above test steps. Move note at end of Test Concept to Notes to Tester. Update test to match formatting of other tests, including updating to the new numbering schema.

Purpose: To verify that the Message Text parameter of generated event notifications is controlled via the Event_Message_Texts_Config property.

Test Concept: Select an object, O1, in the IUT that supports the Event_Message_Texts_Config property. Make O1 perform each supported event transition ~~(i.e., to-offnormal, to-normal, and to-fault)~~. Verify that the 'Message Text' parameter matches the associated Event_Message_Texts_Config value. ~~Note that due to the use of text substitution codes, the resulting text might not be an exact match.~~

Configuration Requirements: If possible, configure each entry in the Event_Message_Texts_Config property of Object O1 to be distinct. ES1 shall be the state to which O1 transitions. D1 is either the pTimeDelay, or pTimeDelayNormal parameter, or 0 (for transitions to and from FAULT state) depending on the event transition. ESINDEX shall be the array index associated with ES1 (1 for offnormal states, 2 for FAULT, and 3 for NORMAL). The notification class for O1 is configured for UnconfirmedEventNotification.

Notes to Tester: For algorithms that do not apply the pTimeDelay or pTimeDelayNormal parameters, assume D1 equals zero when executing the test.

Notes to Tester: Due to the use of text substitution codes, the resulting text might not be an exact match.

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

© 2026 by BACnet International. All rights reserved.

necessary to test both.

Test Steps:
1. MAKE (a condition exist which will cause O1 to transition to ES1)
2. WAIT D1
3. BEFORE **Notification Fail Time**
4. RECEIVE UnconfirmedEventNotification-Request
'Process Identifier' = (any valid process identifier),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = O1,
'Time Stamp' = (any valid time stamp),
'Notification Class' = (the notification class configured for O1),
'Priority' = (any valid priority),
'Event Type' = (any standard event type),
'Message Text' = T1,
'Notify Type' = ALARM | EVENT,
'AckRequired' = TRUE | FALSE,
'From State' = (any valid event state),
'To State' = ES1,
'Event Values' = (any values appropriate to the event type)
~~4.~~*5.* CHECK (T1 is equivalent to Event_Message_Texts_Config[ESINDEX] with any text substitutions as defined by the vendor)

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

7.3.1.19.1 Event_Algorithm_Inhibit Test

Reason For Change: Add new Notes to Tester to indicate how to execute the tests with algorithms that do not support the pTimeDelay or pTimeDelayNormal parameters. Move Notes to Tester from after test to before test. Update test to match formatting of other tests, including updating to the new numbering schema. Fix Typo in Step 8.

Purpose: To verify that Event_Algorithm_Inhibit property in objects with intrinsic or algorithmic reporting controls whether or not the event state detection algorithm is executed.

Test Concept: Select an event generating object, O1, which supports the Event_Algorithm_Inhibit property and does not support the Event_Algorithm_Inhibit_Ref property or where the Event_Algorithm_Inhibit_Ref property is present and not initialized. With Event_Algorithm_Inhibit set to FALSE, make a condition exist that should result in an event transition to a ~~normal~~*NORMAL* or offnormal state. Verify that a transition occurs and that a notification is generated. Set Event_Algorithm_Inhibit to TRUE. If not already in a NORMAL state, verify that the object transitions to NORMAL. Make a condition exist that should result in an event transition, if the object Event_Algorithm_Inhibit were FALSE. If O1 supports fault detection, make a fault condition exist and verify that object detects the fault and transitions to FAULT.

Configuration Requirements: O1 is configured to detect and report unconfirmed events, is in the NORMAL state, and, if supported, is configured to detect fault conditions. D1 is either the pTimeDelay, or pTimeDelayNormal parameter, or 0 (for transitions to and from FAULT state) depending on the event transition.

Notes to Tester: For algorithms that do not apply the pTimeDelay or pTimeDelayNormal parameters, assume D1 equals zero when executing the test.

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

Test Steps:

1. VERIFY pCurrentState = NORMAL
2. VERIFY Event_Algorithm_Inhibit = FALSE
3. MAKE (a condition exist which will result in a transition of O1. If possible, 'To State' shall be an offnormal event state)
4. WAIT D1

© 2026 by BACnet International. All rights reserved.

5. BEFORE **Notification Fail Time**
6. RECEIVE UnconfirmedEventNotification-Request
'Process Identifier' = (~~PID1: any valid process identifier~~*any valid process ID*),
'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' = (ET1: any valid event type),
'Message Text' = (optional, any valid message text),
'Notify Type' = (value from the Notify_Type property configured for O1),
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = (ES1: any event state appropriate to the event type)
'Event Values' = (any values appropriate to the event type)
~~6.~~*7.* WRITE Event_Algorithm_Inhibit = TRUE
~~7.~~*8.* IF (ES1 <> NORMAL) THEN
BEFORE **Notification Fail Time**
RECEIVE UnconfirmedEventNotification-Request
'Process Identifier' = ~~PID1~~*any valid process ID*,
'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' ET1,~~*'Event Type' = ET1*
'Message Text' = (optional, any valid message text),
'Notify Type' = (value from the Notify_Type property configured for O1),
'AckRequired' = TRUE | FALSE,
'From State' = ES1,
'To State' = NORMAL,
'Event Values' = (any values appropriate to the event type)
~~8.~~*9.* VERIFY pCurrentState = NORMAL
~~9.~~*10.* MAKE (a condition exist which would result in a transition of O1 other than TO-FAULT, if Event_Algorithm_Inhibit were FALSE)
~~10.~~*11.* WAIT D1
~~11.~~*12.* WAIT **Notification Fail Time**
~~12.~~*13.* CHECK (that the IUT did not send any event notifications for O1)
~~13.~~*14.* VERIFY pCurrentState = NORMAL
~~14.~~*15.* IF (O1 supports fault detection) THEN {
16. MAKE (a fault condition exist for O1)
17. BEFORE **Notification Fail Time**
18. RECEIVE UnconfirmedEventNotification-Request
'Process Identifier' = ~~PID1~~*any valid process ID,*
'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' = CHANGE_OF_RELIABILITY,
'Message Text' = (optional, any valid message text),
'Notify Type' = (value from the Notify_Type property configured for O1),
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = FAULT,
'Event Values' = (any values appropriate for CHANGE_OF_RELIABILITY)
19. MAKE (remove the fault condition)
20. WAIT (pTimeDelayNormal)
21. BEFORE **Notification Fail Time**
22. RECEIVE UnconfirmedEventNotification-Request

© 2026 by BACnet International. All rights reserved.

'Process Identifier' = ~~PID1~~*any valid process ID*,
'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' = CHANGE_OF_RELIABILITY,
'Message Text' = (optional, any valid message text),
'Notify Type' = (value from the Notify_Type property configured for O1),
'AckRequired' = TRUE | FALSE,
'From State' = FAULT,
'To State' = NORMAL,
'Event Values' = (any values appropriate for CHANGE_OF_RELIABILITY)
}

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

[Modify 7.3.1.22.1 in BTL Specified Tests]

7.3.1.22.1 Event_Detection_Enable Inhibits Event Generation

Reason for Change: Add Notes to Tester to indicate how to handle events that do not support pTimeDelay or pTimeDelayNormal. Move conditionality in original Configuration Requirements to be a separate Test Conditionality. Rewrite Configuration Requirements for better clarity.

Reason for Change: Updated purpose

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 supports event reporting. 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. Make a condition exist that would cause a transition if event reporting were enabled and observe that no notification messages are transmitted.

Test Conditionality: If Event_Detection_Enable cannot be set to FALSE, this test shall be skipped.

~~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). 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.~~
Configuration Requirements: O1 is configured with Event_Detection_Enable set to FALSE. DELAY is the pTimeDelay parameter, pTimeDelayNormal parameter, or zero (for transitions to and from FAULT state) depending on the event transition. NO_TS equals a BACnetDateTime with all unspecified values, a BACnetTime with all unspecified values, or a sequence number of zero.

Notes to Tester: For algorithms that do not apply the pTimeDelay or pTimeDelayNormal parameters, assume DELAY equals zero when executing the test.

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

Test Steps:

1. VERIFY Event_Detection_Enable = FALSE
2. VERIFY pCurrentState = NORMAL
3. VERIFY Acked_Transitions = (T,T,T)
4. VERIFY Event_Time_Stamps = [NO_TS, NO_TS, NO_TS]
5. IF (Event_Message_Texts property exists) THEN

© 2026 by BACnet International. All rights reserved.

6. VERIFY Event_Message_Texts = ['', '', '']
7. MAKE (a condition exist which would cause O1 to transition, if Event_Detection_Enable were TRUE)
8. WAIT (DELAY + **Notification Fail Time)**
9. CHECK (that the IUT did not send any event notifications for O1)
10. VERIFY pCurrentState = NORMAL
11. VERIFY Acked_Transitions = (T,T,T)
12. VERIFY Event_Time_Stamps = [NO_TS, NO_TS, NO_TS]
13. IF (Event_Message_Texts property exists) THEN
14. VERIFY Event_Message_Texts = ['', '', '']

[Modify 7.3.2.21.3.1 in BTL Specified Tests]

7.3.2.21.3.1 ~~ValidDays~~*Valid Days* Test

Reason for Change: Add a new Notes to Tester to account for possibility that the event-generating object's event algorithm does not support the pTimeDelay or pTimeDelayNormal parameters. Add a variable, O1, to represent the event generating object linked to the Notification Class object. Update the test to factor in all event algorithms and update to the new number schema. Move applicable conditionalities in Test Concept to a separate Test Conditionality and reword for clarity.

Reason For Change: Fix delay reference.

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

Test Concept: The TD will select one instance of the Notification Class object and one instance of an event-generating ~~object~~*object, O1,* that is linked to the Notification Class object. The Recipient_List of the Notification Class object shall contain a single recipient with the Valid Days parameter configured so that at least one day is TRUE and at least one day is FALSE. The properties of ~~the event-generating object~~*O1* will be manipulated to cause the Event_State to ~~change from NORMAL to OFFNORMAL.~~*transition*. The tester verifies that if the local date is one of the valid days a notification message is transmitted and if the local date is not a valid day then no notification message is transmitted. ~~For devices that implement a read-only Recipient_List property for all instances of Notification Class objects and are exclusively configured for all days (Valid Days set to all Days), this test shall be omitted. For devices that implement a writeable Recipient_List property for all instances of Notification Class objects, and exclusively accept all days as the only permitted configuration, this test shall be omitted.~~

Test Conditionality: If the Valid Days parameter of the Recipient_List property of the Notification Class object only accepts all days as the only permitted configuration, this test shall be omitted.

Configuration Requirements: The IUT shall be configured with one or more instance of the Notification Class object and at least one event-generating ~~object~~*object, O1,* that is linked to the Notification Class object. ~~The event-generating object~~*O1* may be any object that supports intrinsic reporting or it may be an Event Enrollment object. ~~The event-generating object~~*O1* shall have the Event_Enable property configured to transmit notification messages for all event transitions. ~~The event-generating object~~*O1* shall be configured to be in a NORMAL event state at the start of the test. The Notification Class object shall be configured with a single recipient in the Recipient_List. The Valid Days parameter shall be configured so that at least one day of the week has a value of TRUE and at least one day of the week has a value of FALSE. The Transitions parameter shall be configured for the recipient to receive notifications for all event transitions. *D1 is either the pTimeDelay, or pTimeDelayNormal parameter, or 0) depending on the event transition.*

Notes to Tester: If O1's event algorithm does not support pTimeDelay or pTimeDelayNormal, assume pTimeDelay or pTimeDelayNormal equal zero when executing this test.

Test Steps:

1. (TRANSMIT TimeSynchronization-Request,
   'Time' = (any time within the window defined by From Time and To Time in the BACnetDestination that corresponds to one of the valid days)) |
   (TRANSMIT UTCTimeSynchronization-Request,
   'Time' = (any time within the window defined by From Time and To Time in the BACnetDestination that corresponds to one of the valid days, converted to UTC)) |
   MAKE (the local date and time = (any time within the window defined by From Time and To Time in the BACnetDestination that corresponds to one of the valid days))
2. ~~WAIT (pTimeDelay + **Notification Fail Time**)~~*WAIT (**Internal Processing Fail Time**)*

© 2026 by BACnet International. All rights reserved.

3. VERIFY pCurrentState = NORMAL
4. IF (pMonitoredValue is writable) THEN
5. WRITE pMonitoredValue = (~~a value that is OFFNORMAL~~*a value which will result in the transition of O1*)
ELSE
6. MAKE (pMonitoredValue have ~~a value that is OFFNORMAL~~*a value which will result in the transition of O1*)
~~5.~~*7.* WAIT (~~pTimeDelay~~*D1*)
~~6.~~*8.* BEFORE **Notification Fail Time**
9. RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (~~the event-generating object configured for this test~~*O1*),
'Time Stamp' = (any valid time stamp),
'Notification Class' = (~~the class corresponding to the object being tested~~*the notification class configured for O1*),
'Priority' = (~~the value configured to correspond to a TO-OFFNORMAL transition~~*the value configured for the transition*),
'Event Type' = (any valid event type),
'Message Text' = (optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = ~~OFFNORMAL~~*(ES1: any event state appropriate to the event type),*
'Event Values' = (values appropriate to the event type)
~~7.~~*10.* TRANSMIT BACnet-SimpleACK-PDU
~~8.~~*11.* VERIFY pCurrentState = ~~OFFNORMAL~~*ES1*
~~9.~~*12.* (TRANSMIT TimeSynchronization-Request,
'Time' = (any time within the window defined by From Time and To time in the BACnetDestination that corresponds to one of the invalid days)) |
(TRANSMIT UTCTimeSynchronization-Request,
Time' = (any time within the window defined by From Time and To Time in the BACnetDestination that corresponds to one of the invalid days, converted to UTC)) |
MAKE (the local date and time = (any time within the window defined by From Time and To Time in the BACnetDestination that corresponds to one of the invalid days))
~~10.~~*13.* IF (pMonitoredValue is writable) THEN
14. WRITE pMonitoredValue = (~~a value that is NORMAL~~*a value which will result in the transition of O1*)
ELSE
15. MAKE (pMonitoredValue have ~~a value that is NORMAL~~*a value which will result in the transition of O1*)
~~11.~~*16.* WAIT (~~*pTimeDelayNormal*pTimeDelay~~*D1* + **Notification Fail Time**)
~~12.~~*17.* CHECK (verify that no notification message was transmitted)

7.3.2.21.3.2 ~~FromTime and ToTime Test~~*From Time and To Time Test*

Reason for Change: Add a new Notes to Tester to account for possibility that the event-generating object's event algorithm does not support the pTimeDelay or pTimeDelayNormal parameters. Update test to use new numbering scheme. Move specified conditionalities in Test Concept to be in a separate Test Conditionality and reword for clarity.

Purpose: To verify the operation of the From Time and To Time parameters of a BACnetDestination as used in the Recipient_List property of the Notification Class object.

Test Concept: The case where the local date and time fall within the window defined by the From Time and To Time parameters is covered by the ~~ValidDays~~*Valid Days* test in 7.3.2.21.3.1. This test uses the same IUT configuration and sets the local time to a value that is one of the ~~*ValidDays*~~*Valid Days* but outside of the window defined by the From Time and To Time parameters. The objective is to verify that an event notification message is not transmitted when the event is triggered. ~~For devices that implement a read-only Recipient_List property for all instances of Notification Class objects and are exclusively configured for all times (From Time set to 00:00:00.0, To_Time set to 23:59:59.99), this test shall be omitted. For devices that implement a writeable Notification Class Recipient_List property for all instances of Notification Class objects, and exclusively accept all times as the only permitted configuration, this test shall be omitted.~~

Test Conditionality: If the From Time and To Time parameters of the Recipient_List property of the Notification Class object only support 0:00:00.0 and 23:59:59.99 respectively, this test shall be omitted.

© 2026 by BACnet International. All rights reserved.

~~Configuration Requirements: The configuration requirements are identical to the requirements in 7.3.2.21.3.1.~~
Configuration Requirements: The IUT shall be configured with one or more instance of the Notification Class object and at least one event-generating object, O1, that is linked to the Notification Class object. O1 may be any object that supports intrinsic reporting or it may be an Event Enrollment object. O1 shall have the Event Enable property configured to transmit notification messages for all event transitions. O1 shall be configured to be in a NORMAL event state at the start of the test. The Notification Class object shall be configured with a single recipient in the Recipient_List. The IUT is configured at a time that is one of the Valid Days, but outside the window defined by the From Time and To Time parameters. The Transitions parameter shall be configured for the recipient to receive notifications for all event transitions. D1 is either the pTimeDelay, or pTimeDelayNormal parameter, or 0 depending on the event transition.

Notes to Tester: If the event generating object's event algorithm does not support pTimeDelay or pTimeDelayNormal, assume pTimeDelay or pTimeDelayNormal equal zero when executing this test.

Test Steps:
1. (TRANSMIT TimeSynchronization-Request,
   'Time' = (any time outside the window defined by From Time and To Time in the BACnetDestination that corresponds to one of the valid days)) |
   (TRANSMIT UTCTimeSynchronization-Request,
   'Time' = (any time outside the window defined by From Time and To Time in the BACnetDestination that corresponds to one of the valid days, converted to UTC)) |
   MAKE (the local date and time = (any time outside the window defined by From Time and To Time in the BACnetDestination that corresponds to one of the valid days))
2. WAIT (~~pTimeDelay~~*D1* + **Notification Fail Time**)
3. VERIFY pCurrentState = NORMAL
4. IF (pMonitoredValue is writable) THEN
   WRITE pMonitoredValue = (~~a value that is OFFNORMAL~~*a value which will result in the transition of the event generating object*)
   ELSE
5. MAKE (pMonitoredValue have ~~a value that is OFFNORMAL~~*a value which will result in the transition of the event generating object)*
~~5.6.~~ WAIT (~~pTimeDelay~~*D1* + **Notification Fail Time**)
~~6.~~*7.* CHECK (verify that no notification message was transmitted)

7.3.2.21.3.5 Recipient_List Property Supports Device Identifier Recipients Test

Reason for Change: Add a new Notes to Tester to account for possibility that the event-generating object's event algorithm does not support the pTimeDelay or pTimeDelayNormal parameters. Move the existing Notes to Tester from after the Test Steps to before the Test Steps. Update to the new numbering schema. Add missing space after Configuration Requirements colon. Remove test conditionality regarding a read only Recipient_List property because this test is executed when Supports Writable Recipient_List is checked.

Purpose: To verify that the Recipient_List property of the Notification Class object supports the device form of the Recipient component and that the IUT is able to associate a MAC address with the Device Identifier. The intent is to ensure that the IUT is able to locate the specified alarm recipient and send notification to the specified recipient. ~~This test shall be run if the IUT's Notification Class object's Recipient_List property supports the BACnet object identifier form of BACnetRecipient.~~

Test Concept: The tester shall select a single event generating object E in the IUT that references Notification Class object N. The tester shall add an entry into the Recipient_List of the associated Notification Class object that specifies a Device Identifier, D, for a device that the IUT is not already aware of. The TD, acting as device D, shall be located on a different network than the IUT to ensure that the IUT is capable of binding to recipients located on any network. ~~For devices that implement a read-only Recipient_List property for all instances of Notification Class objects, and there is an address form of the Recipient component in all instances, this test shall be omitted.~~ D1 is either the pTimeDelay, or pTimeDelayNormal parameter, or 0 (for transitions to and from FAULT state) depending on the event transition.

Configuration ~~Requirements:The~~*Requirements: The* TD shall be configured so that it does not execute WhoHas.

Notes to Tester: If the event generating object's event algorithm does not support pTimeDelay or pTimeDelayNormal, assume pTimeDelay or pTimeDelayNormal equal zero when executing this test.

© 2026 by BACnet International. All rights reserved.

Notes to Tester: The IUT is expected to initiate one or more range-restricted WhoIs requests after the modification of the Recipient_List but before the sending of the notification. The IUT might also need to perform other network discovery operations. Given that there are multiple approaches to the use of WhoIs for device discovery, the test only focuses on the IUT's ability to find device D and not on the specifics or timing of the WhoIs requests.

Test Steps:

1. WRITE N.RecipientList = ({all days, all times, D, any process ID, FALSE, all transitions})
2. MAKE (a condition exist that will cause E to generate an event transition)
3. WAIT D1
4. BEFORE (**Notification Fail Time** plus the amount of time the IUT takes to perform device discovery)
5. RECEIVE UnconfirmedEventNotification-Request,
   'Process Identifier' = ~~PID~~*any valid process ID*,
   'Initiating Device Identifier' = IUT,
   'Event Object Identifier' = E,
   'Time Stamp' = (any valid time stamp),
   'Notification Class' = (N's instance),
   'Priority' = (any valid priority),
   'Event Type' = (any valid event type),
   'Message Text' = (optional, any valid message text),
   'Notify Type' = ALARM | EVENT,
   'AckRequired' = TRUE | FALSE,
   'From State' = (any valid event state),
   'To State' = (any valid event state),
   'Event Values' = (values appropriate to the event type)

~~Notes to Tester: The IUT is expected to initiate one or more range-restricted WhoIs requests after the modification of the Recipient_List but before the sending of the notification. The IUT might also need to perform other network discovery operations. Given that there are multiple approaches to the use of WhoIs for device discovery, the test only focuses on the IUT's ability to find device D and not on the specifics or timing of the WhoIs requests.~~

7.3.2.21.3.6 Recipient_List Property Supports Network Address Recipients

Reason for Change: Add a new Notes to Tester to account for possibility that the event-generating object's event algorithm does not support the pTimeDelay or pTimeDelayNormal parameters. Update test to new numbering schema. Remove test conditionality regarding a read only Recipient_List property because this test is executed when Supports Writable Recipient_List is checked.

Purpose: To verify that the Recipient_List property of the Notification Class object supports the address form of the Recipient component. The intent is to ensure that the IUT is able to send notifications to the specified recipient.

Test Concept: The tester shall select a single event-generating object E in the IUT that references Notification Class object N. The tester shall add an entry into the Recipient_List of the associated Notification Class object that specifies a BACnetAddress A, where A is a unicast or is a local, remote, or global broadcast address. ~~For devices that implement a readonly Recipient_List property for all instances of Notification Class objects, and there is a Device Identifier form of the Recipient component in all instances, this test shall be skipped.~~ D1 is either the pTimeDelay, or pTimeDelayNormal parameter, or 0 (for transitions to and from FAULT state) depending on the event transition.

Notes to Tester: If the event generating object's event algorithm does not support pTimeDelay or pTimeDelayNormal, assume pTimeDelay or pTimeDelayNormal equal zero when executing this test.

Test Steps:
1. WRITE N.RecipientList = ({all days, all times, A, any process ID, FALSE, all transitions})
2. MAKE (a condition exist that will cause E to generate an event transition)
3. WAIT D1
4. BEFORE **Notification Fail Time**
5. RECEIVE
   DESTINATION = A,
   UnconfirmedEventNotification-Request,

© 2026 by BACnet International. All rights reserved.

'Process Identifier' = ~~PID~~*any valid process ID*,
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = E,
'Time Stamp' = (any valid time stamp),
'Notification Class' = (N's instance),
'Priority' = (any valid priority),
'Event Type' = (any valid event type),
'Message Text' = (optional, any valid message text),
'Notify Type' = ALARM | EVENT,
'AckRequired' = TRUE | FALSE,
'From State' = (any valid event state),
'To State' = (any valid event state),
'Event Values' = (values appropriate to the event type)

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp4-4: Dotted Decimal Host Name Test Updates [BTLWG-1860]

Overview:

An IR at the SSPC confirmed that the dotted decimal form of host name must be supported in FD_BBMD_Address.

Changes:

Checklist Changes

None

Test Plan Changes

9.3.3 Is Able to Operate in Foreign Mode

The IUT can register as a foreign device with a BBMD.

The IUT supports a configurable BBMD Address to which it sends Register-Foreign-Device NPDU.

...		
135.1-2025 - 12.3.8.7 - Register-Foreign-Device when NPOs Supported		
	Test Conditionality	Must be executed if the IUT claims Protocol_Revision >= 17.
	Test Directives	Repeat this test with the IUT configured to register as a foreign device with the TD that is using a valid BBMD_Address,~~ once using the host ip-address and again using the host name.~~ once using the host ip-address choice, once using the host name choice in dotted-decimal ip address form, and once using the host name choice in domain name form.
	Testing Hints	
...		

Specified Test Changes

None

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp4-5: Change of State Tests Require Local Date and Time [BTLWG-1861]

Overview:

The change of state tracking properties requires the Device support Local_Date and Local_Time properties.

Changes:

Checklist Changes

None

Test Plan Changes

3.5 Binary Input Object

3.5.4 Supports Change Of State Tracking

The IUT contains or can be made to contain an object with the Change_Of_State_Time, Change_Of_State_Count and Time_Of_State_Count_Reset properties.

...		
Verify EPICS		
	Test Conditionality	*Must be executed.*
	Test Directives	*Verify in the EPICS that the Local_Time and Local_Date properties of the Device object are present.*
	Testing Hints	

3.6 Binary Output Object

3.6.5 Supports Change Of State Tracking

The IUT contains or can be made to contain and object with the Change_Of_State_Time, Change_Of_State_Count and Time_Of_State_Count_Reset properties.

...		
Verify EPICS		
	Test Conditionality	*Must be executed.*
	Test Directives	*Verify in the EPICS that the Local_Time and Local_Date properties of the Device object are present.*
	Testing Hints	

3.7 Binary Value Object

3.7.3 Supports Change Of State Tracking

The IUT contains or can be made to contain an object with the Change_Of_State_Time, Change_Of_State_Count and Time_Of_State_Count_Reset properties.

...		
Verify EPICS		
	Test Conditionality	*Must be executed.*
	Test Directives	*Verify in the EPICS that the Local_Time and Local_Date properties of the Device object are present.*

© 2026 by BACnet International. All rights reserved.

	Testing Hints	

Specified Test Changes

None

© 2026 by BACnet International. All rights reserved.

BTL-26.1 imp4-6: Notification Class Transition Test [BTLWG-1859]

Overview:

7.3.2.21.3.4 Transitions Test does not account for event algorithms that only support Normal-to-Normal transitions.

Changes:

Checklist Changes

None

Test Plan Changes

[Modify the Notification Class Object section Base Requirements]
[Modify the Notification Class Object section Supports Writable Recipient_List Property]

3.17.1 Base Requirements

Base requirements must be met by any IUT that can contain Notification Class objects.

…		
~~**135.1-2025 - 7.3.2.21.3.4 - Transitions Test**~~		
	~~**Test Conditionality**~~	~~Must be executed.~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	
135.1-2025 - 7.3.2.21.3.7 - Recipient_List non-volatility Test		
	Test Conditionality	Must be executed if the IUT claims Protocol_Revision 13 or greater.
	Test Directives	Repeat the test, via ReinitializeDevice command if supported, and with a power interruption.
	Testing Hints	

3.17.3 Supports Writable Recipient_List Property

The IUT supports Recipient_List properties that are modifiable via write services. All required values must be accepted in write requests.

…		
135.1-2025 - 9.23.2.20 - Time Non-Pattern Properties Test using WritePropertyMultiple Service		
	Test Conditionality	This test shall only be applied to devices claiming Protocol_Revision 11 or higher and which supports execution of WritePropertyMultiple.
	Test Directives	Apply to the fromTime and again to the toTime element in a Recipient_List property in a Notification Class.
	Testing Hints	
BTL - 7.3.2.21.3.X1 - Transitions Test Normal-to-Offnormal		
	Test Conditionality	*Must be executed if the IUT supports normal to offnormal event generating objects and the IUT does not exclusively accept all transitions as the only permitted configuration for the Transitions parameter.*
	Test Directives	
	Testing Hints	
BTL - 7.3.2.21.3.X2 - Transitions Test Normal-to-Fault Test		
	Test Conditionality	*Must be executed if the IUT supports normal to fault event generating objects and the IUT does not exclusively accept all transitions as the only permitted configuration for the Transitions parameter.*
	Test Directives	
	Testing Hints	

© 2026 by BACnet International. All rights reserved.

BTL - 7.3.2.21.3.X3 - Transitions Test Normal-to-Normal Test		
	Test Conditionality	*Must be executed if the IUT supports normal to normal event generating objects and the IUT does not exclusively accept all transitions as the only permitted configuration for the Transitions parameter.*
	Test Directives	
	Testing Hints	

Specified Test Changes

[Move test 7.3.2.21.3.4 from 135.1-2025 into BTL Specified Tests and remove the test]

7.3.2.21.3.4 Transitions Test

Reason for Change: Remove and replace with 7.3.2.21.3.X1, 7.3.2.21.3.X2, and 7.3.2.21.3.X3.

~~Purpose: To verify that notification messages are transmitted only if the bit in the Transitions parameter corresponding to the event transition is set.~~

~~Test Concept: The IUT is configured such that the Transitions parameter indicates that some event transitions are to trigger an event notification and some are not. Each event transition is triggered and the IUT is monitored to verify that notification messages are transmitted only for those transitions for which the Transitions parameter has a value of TRUE. For devices that implement a read-only Recipient_List property for all instances of Notification Class objects, and are exclusively configured for all transitions (all bits in Transitions set to TRUE), this test shall be omitted. For devices that implement a writeable Notification Class Recipient_List property for all instances of Notification Class objects, and exclusively accept all transitions as the only permitted configuration, this test shall be omitted.~~

~~Configuration Requirements: The IUT shall be configured with one or more instance of the Notification Class object and at least one event-generating object that is linked to the Notification Class object. The event-generating object shall have the Event_Enable property configured to transmit notification messages for all event transitions. The event-generating object shall be configured to be in a NORMAL event state at the start of the test. The Notification Class object shall be configured with a single recipient in the Recipient_List. The Transitions parameter shall be configured with a value of TRUE for either the TO-OFFNORMAL transition or the TO-NORMAL transition and the other event transition shall have a value of FALSE. The local time shall be configured such that it represents one of the valid days in the window specified by From Time and To Time.~~

~~Test Steps:~~

~~1. VERIFY pCurrentState = NORMAL~~
~~2. WAIT (pTimeDelay + **Notification Fail Time**)~~
~~3. IF (pMonitoredValue is writable) THEN~~
~~WRITE pMonitoredValue = (a value that is OFFNORMAL)~~
~~ELSE~~
~~MAKE (pMonitoredValue have a value that is OFFNORMAL)~~
~~4. WAIT (pTimeDelay)~~
~~5. BEFORE **Notification Fail Time**~~
~~IF (the Transitions bit corresponding to the TO-OFFNORMAL*TO_OFFNORMAL* transition is TRUE) THEN~~
~~RECEIVE ConfirmedEventNotification Request,~~
~~'Process Identifier' = (any valid process ID),~~
~~'Initiating Device Identifier' = IUT,~~
~~'Event Object Identifier' = (the event-generating object configured for this test),~~
~~'Time Stamp' = (any valid time stamp),~~
~~'Notification Class' = (the class corresponding to the object being tested),~~
~~'Priority' = (the value configured to correspond to a TO-OFFNORMAL transition),~~
~~'Event Type' = (any valid event type),~~
~~'Notify Type' = EVENT | ALARM,~~
~~'AckRequired' = TRUE | FALSE,~~
~~'From State' = NORMAL,~~
~~'To State' = OFFNORMAL,~~

© 2026 by BACnet International. All rights reserved.

'Event Values' = (values appropriate to the event type)
TRANSMIT BACnet-SimpleACK-PDU
ELSE
CHECK (verify that the IUT did not transmit an event notification message)
6. VERIFY pCurrentState = OFFNORMAL
7. IF (pMonitoredValue is writable) THEN
WRITE pMonitoredValue = (a value that is NORMAL)
ELSE
MAKE (pMonitoredValue have a value that is NORMAL)
8. WAIT (pTimeDelayNormal)
9. BEFORE **Notification Fail Time**
IF (the Transitions bit corresponding to the TO-NORMAL transition is TRUE) THEN
RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the event-generating object configured for this test),
'Time Stamp' = (any valid time stamp),
'Notification Class' = (the class corresponding to the object being tested),
'Priority' = (the value configured to correspond to a TO-NORMAL transition),
'Event Type' = (any valid event type),
'Message Text' = (optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = OFFNORMAL,
'To State' = NORMAL,
'Event Values' = (values appropriate to the event type)
TRANSMIT BACnet-SimpleACK-PDU
ELSE
CHECK (verify that the IUT did not transmit an event notification message)
10. VERIFY pCurrentState = NORMAL
11. IF (the event-triggering object can be placed into a fault condition) THEN {
MAKE (a condition exist that will cause the object to generate a TO_FAULT transition)
BEFORE **Notification Fail Time**
IF (the Transitions bit corresponding to the TO-FAULT transition is TRUE) THEN
RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the event-generating object configured for this test),
'Time Stamp' = (any valid time stamp),
'Notification Class' = (the class corresponding to the object being tested),
'Priority' = (the value configured to correspond to a TO-FAULT transition),
'Event Type' = (IF (Protocol_Revision < 13) THEN
(any valid event type),
ELSE
CHANGE_OF_RELIABILITY),
'Message Text' = (optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = FAULT,
'Event Values' = (values appropriate to the event type)
TRANSMIT BACnet-SimpleACK-PDU
ELSE
CHECK (verify that the IUT did not transmit an event notification message)
VERIFY pCurrentState = FAULT
}

Notes to Tester: The UnconfirmedEventNotification service may be substituted for the ConfirmedEventNotification service, in which case the TD shall skip all of the steps in which a BACnet-SimpleACK-PDU is sent.

© 2026 by BACnet International. All rights reserved.

[Add new test, 7.3.2.21.3.X1, to BTL Specified Tests]

7.3.2.21.3.X1 Transitions Test Normal-to-Offnormal

Reason for Change: No test exists for this functionality.

Purpose: To verify that notification messages are transmitted only if the bit in the Transitions parameter corresponding to the event transition is set for TO_OFFNORMAL or TO_NORMAL event transitions.

Test Concept: The IUT shall be configured such that TO_OFFNORMAL event transitions generate event notifications and TO_NORMAL event transitions do not generate notifications. TO_OFFNORMAL and TO_NORMAL event transitions shall be triggered, and the IUT shall be monitored to verify that notification messages are transmitted only for TO_OFFNORMAL transitions. The test shall then be repeated with the TO_OFFNORMAL and TO_NORMAL Transition settings reversed.

Test Conditionality: This test shall be skipped if the IUT does not contain a normal to offnormal event generating object or the IUT exclusively accepts all transitions as the only permitted configuration for the Transitions parameter of the Notification Class Recipient_List property.

Configuration Requirements: The IUT shall be configured with one or more instance of the Notification Class object and at least one event-generating object that is linked to the Notification Class object under test, NC1. The event-generating object, O1, may be any normal to offnormal event generating object. O1 shall have the Event_Enable property configured to transmit messages for all event transitions. O1 shall be configured to be in a NORMAL event state at the start of the test. NC1 shall be configured with a single recipient, R, in the Recipient_List property with the Transitions parameter set to (TRUE, FALSE, FALSE). The local date and time shall be configured such that the current day is included in VD, where VD is the ValidDays value, and the current time falls within TW, where TW is the time window defined by the From Time and To Time values.

Notes to Tester: For algorithms that do not apply the pTimeDelay or pTimeDelayNormal parameters, assume pTimeDelay or pTimeDelayNormal equals zero when executing this test.

Notes to Tester: The UnconfirmedEventNotification service may be substituted for the ConfirmedEventNotification service. When executing the test using the UnconfirmedEventNotification service, the value written for the Recipient_List's Issue Confirmed Notification parameter shall be FALSE instead of TRUE. The TD shall also skip all steps requiring transmission of a BACnet-SimpleACK-PDU.

Test Steps:

1. VERIFY pCurrentState = NORMAL
2. WAIT (pTimeDelay + **Notification Fail Time**)
3. IF (pMonitoredValue is writable) THEN
4. WRITE pMonitoredValue = (a value that is OFFNORMAL)
   ELSE
5. MAKE (pMonitoredValue have a value that is OFFNORMAL)
6. WAIT (pTimeDelay)
7. BEFORE **Notification Fail Time**
8. RECEIVE ConfirmedEventNotification-Request
   'Process Identifier' = (any valid process ID)
   'Initiating Device Identifier' = IUT,
   'Event Object Identifier' = (O1)
   'Time Stamp' = (any valid time stamp),
   'Notification Class' = (NC1)
   'Priority' = (the value configured to correspond to a TO_OFFNORMAL transition),
   'Event Type' = (any valid event type),
   'Notify Type' = EVENT | ALARM,
   'AckRequired' = TRUE | FALSE,
   'From State' = NORMAL,
   'To State' = OFFNORMAL,
   'Event Values' = (values appropriate to the event type)
9. TRANSMIT BACnet-SimpleACK-PDU
10. VERIFY pCurrentState = OFFNORMAL
11. IF (pMonitoredValue is writable) THEN

© 2026 by BACnet International. All rights reserved.

12. WRITE pMonitoredValue = (a value that is NORMAL)
    ELSE
13. MAKE (pMonitoredValue have a value that is NORMAL)
14. WAIT (pTimeDelayNormal + **Notification Fail Time**)
15. CHECK (verify that the IUT did not transmit an event notification message)
16. VERIFY pCurrentState = NORMAL
17. WRITE NC1.RecipientList = ({VD, TW, R, PID1, TRUE, (FALSE, FALSE, TRUE)})
18. IF (pMonitoredValue is writable) THEN
19. WRITE pMonitoredValue = (a value that is OFFNORMAL)
    ELSE
20. MAKE (pMonitoredValue have a value that is OFFNORMAL)
21. WAIT (pTimeDelay + **Notification Fail Time**)
22. CHECK (verify that the IUT did not transmit an event notification message)
23. VERIFY pCurrentState = OFFNORMAL
24. IF (pMonitoredValue is writable) THEN
25. WRITE pMonitoredValue = (a value that is NOMRAL)
    ELSE
26. MAKE (pMonitoredValue have a value that is NORMAL)
27. WAIT (pTimeDelayNormal)
28. BEFORE **Notification Fail Time**
29. RECEIVE ConfirmedEventNotification-Request
        'Process Identifier' = (PID1)
        Initiating Device Identifier' = IUT,
        'Event Object Identifier' = (O1)
        'Time Stamp' = (any valid time stamp),
        'Notification Class' = (NC1)
        'Priority' = (the value configured to correspond to a TO_NORMAL transition),
        'Event Type' = (any valid event type),
        'Notify Type' = EVENT | ALARM,
        'AckRequired' = TRUE | FALSE,
        'From State' = OFFNORMAL,
        'To State' = NORMAL,
        'Event Values' = (values appropriate to the event type)
30. TRANSMIT BACnet-SimpleACK-PDU
31. VERIFY pCurrentState = NORMAL

[Add new test, 7.3.2.21.3.X2, to BTL Specified Tests]

7.3.2.21.3.X2 Transitions Test Normal-to-Fault

Reason for Change: No test exists for this functionality.

Purpose: To verify that notification messages are transmitted only if the bit in the Transition parameter corresponding to the event transitions is set for TO_FAULT or TO_NORMAL event transitions.

Test Concept: The IUT shall be configured such that TO_FAULT event transitions generate event notifications and TO_NORMAL event transitions do not generate notifications. TO_FAULT and TO_NORMAL event transitions shall be triggered, and the IUT shall be monitored to verify that notification messages are transmitted only for TO_FAULT transitions. The test shall then be repeated with the TO_FAULT and TO_NORMAL Transition settings reversed.

Test Conditionality: This test shall be skipped if the IUT does not contain a normal to fault event generating object or the IUT exclusively accepts all transitions as the only permitted configuration for the Transitions parameter of the Notification Class Recipient_List property.

Configuration Requirements: The IUT shall be configured with one or more instance of the Notification Class object and at least one event-generating object that is linked to the Notification Class object under test, NC1. The event-generating object, O1, may be any normal to fault event generating object. O1 shall have the Event_Enable property configured to transmit messages for all event transitions. O1 shall be configured to be in a NORMAL event state at the start of the test. NC1 shall be configured with a single recipient in the Recipient_List property with the Transitions parameter set to (FALSE, TRUE,

© 2026 by BACnet International. All rights reserved.

FALSE). The local date and time shall be configured such that the current day matches a day specified by the ValidDays parameter and the current time is between the From Time and To Time parameters.

Notes to Tester: The UnconfirmedEventNotification service may be substituted for the ConfirmedEventNotification service. When executing the test using the UnconfirmedEventNotification service, the value written for the Recipient_List's Issue Confirmed Notification parameter shall be FALSE instead of TRUE. The TD shall also skip all steps requiring transmission of a BACnet-SimpleACK-PDU.

Test Steps:

1. VERIFY pCurrentState = NORMAL
2. WAIT (pTimeDelay + **Notification Fail Time)**
3. MAKE (a condition exist that will cause the object to generate a TO_FAULT transition)
4. BEFORE **Notification Fail Time**
5. RECEIVE ConfirmedEventNotification-Request
   'Process Identifier' = (any valid process ID)
   Initiating Device Identifier' = IUT,
   'Event Object Identifier' = (O1)
   'Time Stamp' = (any valid time stamp),
   'Notification Class' = (NC1)
   'Priority' = (the value configured to correspond to a TO_FAULT transition),
   'Event Type' = (any valid event type),
   'Notify Type' = EVENT | ALARM,
   'AckRequired' = TRUE | FALSE,
   'From State' = NORMAL,
   'To State' = FAULT,
   'Event Values' = (values appropriate to the event type)
6. TRANSMIT BACnet-SimpleACK-PDU
7. VERIFY pCurrentState = FAULT
8. MAKE (a condition exist that will cause the object to generate a TO_NORMAL transition)
9. WAIT (pTimeDelayNormal + **Notification Fail Time)**
10. CHECK (verify that the IUT did not transmit an event notification message)
11. WRITE NC1.RecipientList = ({VD, TW, R, PID1, TRUE, (FALSE, FALSE, TRUE)})
12. MAKE (a condition exist that will cause the object to generate a TO_FAULT transition)
13. WAIT **Notification Fail Time**
14. CHECK (verify that the IUT did not transmit an event notification message)
15. MAKE (a condition exist that will cause the object to generate a TO_NORMAL transition)
16. WAIT (pTimeDelayNormal)
17 BEFORE **Notification Fail Time**
18. RECEIVE ConfirmedEventNotification-Request
   'Process Identifier' = (PID1)
   Initiating Device Identifier' = IUT,
   'Event Object Identifier' = (O1)
   'Time Stamp' = (any valid time stamp),
   'Notification Class' = (NC1)
   'Priority' = (the value configured to correspond to a TO_NORMAL transition),
   'Event Type' = (any valid event type),
   'Notify Type' = EVENT | ALARM,
   'AckRequired' = TRUE | FALSE,
   'From State' = FAULT,
   'To State' = NORMAL,
   'Event Values' = (values appropriate to the event type)
19. TRANSMIT BACnet-SimpleACK-PDU
20. VERIFY pCurrentState = NORMAL

[Add new test, 7.3.2.21.3.X3, to BTL Specified Tests]

7.3.2.21.3.X3 Transitions Test Normal-to-Normal

© 2026 by BACnet International. All rights reserved.

Reason for Change: No test exists for this functionality.

Purpose: To verify that notification messages are transmitted only if the bit in the Transitions parameter corresponding to the event transition is set for TO_NORMAL event transitions.

Test Concept: The IUT shall be configured such that TO_NORMAL event transitions do not generate notifications. A TO_NORMAL event transitions shall be triggered, and the IUT shall be monitored to verify that notification messages are not transmitted. The Recipient_List's Transitions parameter shall then be written, such that TO_NORMAL event transitions trigger a notification message to be transmitted.

Test Conditionality: This test shall be skipped if the IUT does not contain a normal-to-normal event generating object or the IUT exclusively accepts all transitions as the only permitted configuration for the Transitions parameter of the Notification Class Recipient_List property.

Configuration Requirements: The IUT shall be configured with one or more instance of the Notification Class object and at least one event-generating object that is linked to the Notification Class object under test, NC1. The event-generating object, O1, may be any normal-to-normal event generating object. O1 shall have the Event_Enable property configured to transmit messages for all event transitions. O1 shall be configured to be in a NORMAL event state at the start of the test. NC1 shall be configured with a single recipient, R, in the Recipient_List property with the Transitions parameter set to (FALSE, FALSE, FALSE). The local date and time shall be configured such that the current day is included in VD, where VD is the ValidDays value, and the current time falls within TW, where TW is the time window defined by the From Time and To Time values.

Notes to Tester: For algorithms that do not apply the pTimeDelay or pTimeDelayNormal parameters, assume pTimeDelay or pTimeDelayNormal equals zero when executing this test.

Notes to Tester: The UnconfirmedEventNotification service may be substituted for the ConfirmedEventNotification service. When executing the test using the UnconfirmedEventNotification service, the value written for the Recipient_List's Issue Confirmed Notification parameter shall be FALSE instead of TRUE. The TD shall also skip all steps requiring transmission of a BACnet-SimpleACK-PDU.

Test Steps:

1. VERIFY pCurrentState = NORMAL
2. WAIT (pTimeDelay + **Notification Fail Time**)
3. IF (pMonitoredValue is writable) THEN
4. WRITE pMonitoredValue = (a value that will result in a TO_NORMAL transition)
   ELSE
5. MAKE (pMonitoredValue have a value that will result in a TO_NORMAL transition)
6. WAIT (pTimeDelayNormal + **Notification Fail Time**)
7. CHECK (verify that the IUT did not transmit an event notification message)
8. VERIFY pCurrentState = NORMAL
9. WRITE NC1.RecipientList = ({VD, TW, R, PID1, TRUE, (FALSE, FALSE, TRUE)})
10. IF (pMonitoredValue is writable) THEN
11. WRITE pMonitoredValue = (a value that will result in a TO_NORMAL transition)
    ELSE
12. MAKE (pMonitoredValue have a value that will result in a TO_NORMAL transition)
13. WAIT (pTimeDelayNormal)
14. BEFORE **Notification Fail Time**
15. RECEIVE ConfirmedEventNotification-Request
    'Process Identifier' = (PID1)
    Initiating Device Identifier' = IUT,
    'Event Object Identifier' = (O1)
    'Time Stamp' = (any valid time stamp),
    'Notification Class' = (NC1)
    'Priority' = (the value configured to correspond to a TO_NORMAL transition),
    'Event Type' = (any valid event type),
    'Notify Type' = EVENT | ALARM,
    'AckRequired' = TRUE | FALSE,
    'From State' = NORMAL,
    'To State' = NORMAL,

© 2026 by BACnet International. All rights reserved.

'Event Values' = (values appropriate to the event type)
16. TRANSMIT BACnet-SimpleACK-PDU
17. VERIFY pCurrentState = NORMAL

© 2026 by BACnet International. All rights reserved.