

Clarification Request

References: BTL Test Plan 9.0.final tests 7.3.2.26.X2 and ...X4

Date of BTL-WG Response: Aug 9, 2012

Background:

BTL specified tests 7.3.2.26.X2, 7.3.2.26.X4

In step 1 of test 7.3.2.26.X4 the property enable of a EventLog Object is written with the value TRUE. In step 2 immediatly after this an event is send to the IUT which is supposed to be stored in the EventLog.

There should be at least processing fail time allowed before sending the event in step 3. The IUT may need time to do internal preparations before beeing ready to process events. The IUT may even try to issue add_list_element requests to recipient lists of external notification_class objects first.

Question:

Should a vendor defined wait be allowed between steps 2 and 3?

Response:

Yes. The tests will be modified as below.

7.3.2.26.X2 Remote Logging of Notifications

Purpose: To verify that the IUT correctly collects and represents the Notifications which it receives.

Test Concept: Make TD send multiple event notification messages. Use ReadRange to retrieve the events from an Event Log or perhaps from multiple Event Logs in the IUT, and compare the two representations.

Configuration Requirements: LO₁ is an Event Log object in IUT which logs the Event types which are sent. Stop_When_Full in LO₁ shall be FALSE or absent.

Test Steps:

1. WRITE Enable = TRUE
2. *WAIT (Vendor Specified Delay)*
3. TRANSMIT ConfirmedEventNotification-Request,
 'Process Identifier' = (any valid process identifier),

'Initiating Device Identifier' = TD,
 'Event Object Identifier' = (any valid object identifier),
 'Time Stamp' = (T1, any valid timestamp),
 'Notification Class' = (any valid notification class),
 'Priority' = (any valid priority),
 'Event Type' = (any standard event type),
 'Message Text' = (any character string),
 'Notify Type' = ALARM | EVENT,
 'AckRequired' = TRUE | FALSE,
 'From State' = (state S1, any valid state for this event type),
 'To State' = (state S2, any valid state for this event type that can follow
 S1),

'Event Values' = (any values appropriate to the event type)

43. RECEIVE BACnet-SimpleACK-PDU

54. TRANSMIT ConfirmedEventNotification-Request,

SOURCE = IUT,
 DESTINATION = TD,
 'Process Identifier' = (any valid process identifier),
 'Initiating Device Identifier' = IUT,
 'Event Object Identifier' = (any valid object identifier),
 'Time Stamp' = (T2, any valid timestamp),
 'Notification Class' = (any valid notification class),
 'Priority' = (any valid priority),
 'Event Type' = (any standard event type),
 'Message Text' = (any character string),
 'Notify Type' = ALARM | EVENT,
 'AckRequired' = TRUE | FALSE,
 'From State' = (state S1, any valid state for this event type),
 'To State' = (state S2, any valid state for this event type that can follow
 S1),

'Event Values' = (any values appropriate to the event type)

65. RECEIVE BACnet-SimpleACK-PDU

76. READ RC = LO1, Record_Count

87. TRANSMIT ReadRange-Request,

'Object Identifier' = LO1,
 'Property Identifier' = Log_Buffer,
 'Reference Index' = RC,
 'Count' = -2

98. RECEIVE ReadRange-ACK,

'Object Identifier' = LO1,
 'Property Identifier' = Log_Buffer,
 'Result Flags' = {FALSE, ?, TRUE },
 'Item Count' = 2,
 'Item Data' = (logged data that matches the information received in steps 3 and 6,
 except that Process_Identifier can be any value and need not match)

109. CHECK (that the events were logged in the order in which they were received)

Notes to Tester: When the UnconfirmedEventNotification service is used instead of the ConfirmedEventNotification service, the test shall skip the steps in which a SimpleACK-PDU is expected.

7.3.2.26.X4 Remote Logging of ACK_NOTIFICATION

Purpose: To verify that the IUT correctly collects and represents ACK_NOTIFICATIONs which it receives.

Test Concept: Send an ACK_NOTIFICATION to the IUT. Use ReadRange to retrieve that same event from an Event Log and compare the two representations.

Configuration Requirements: LO₁ is an Event Log object in IUT which logs ACK_NOTIFICATIONs. Stop_When_Full in LO₁ shall be FALSE or absent.

Test Steps:

1. WRITE Enable = TRUE

2. *WAIT (Vendor Specified Delay)*

32. TRANSMIT ConfirmedEventNotification-Request,
 SOURCE = IUT,
 DESTINATION = TD,
 'Process Identifier' = (any valid process identifier),
 'Initiating Device Identifier' = IUT,
 'Event Object Identifier' = (any valid object identifier),
 'Time Stamp' = (T1, any valid timestamp),
 'Notification Class' = (any valid notification class),
 'Priority' = (any valid priority),
 'Event Type' = (any standard event type),
 'Message Text' = (any character string),
 'Notify Type' = ACK_NOTIFICATION,
 'From State' = (state S1, any valid state for this event type)

43. RECEIVE BACnet-SimpleACK-PDU

54. READ RC = LO1, Record_Count

65. TRANSMIT ReadRange-Request,
 'Object Identifier' = LO1,
 'Property Identifier' = Log_Buffer,
 'Reference Index' = RC,
 'Count' = -1

76. RECEIVE ReadRange-ACK,
 'Object Identifier' = LO1,
 'Property Identifier' = Log_Buffer,
 'Result Flags' = {FALSE, ?, TRUE },
 'Item Count' = 1,
 'Item Data' = (logged data that matches the information received in step 2,
 except that Process_Identifier can be any value and need not match)

Notes to Tester: When the UnconfirmedEventNotification service is used instead of the ConfirmedEventNotification service, the test shall skip the step in which a SimpleACK-PDU is expected.