

Clarification Request

References: BTL Specified Tests-26.0, BTL TP 23.3 Addendum imp2, test '7.3.1.11.X2'

Date of BTL-WG Response: April 3, 2025

Background: BTL Specified Tests-26.0, BTL TP 23.3 Addendum imp2, test '7.3.1.11.X2'

7.3.1.11.X2 Acked_Transitions Test for Normal to Fault Transitions

Reason for Change: New test.

Purpose: To verify that the Acked_Transitions property tracks whether or not an acknowledgment has been received for a previously issued fault 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 fault event transitions are to trigger an event notification. The Acked_Transitions property shall have the value (?, TRUE, ?). The fault event transition is triggered and the Acked_Transitions property is monitored to verify that the appropriate bit is cleared when a notification message is transmitted and reset when an acknowledgment is received.

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

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

Test Steps:

1. VERIFY pCurrentState = NORMAL
2. VERIFY Acked_Transitions = (?, TRUE, ?)
3. IF (Protocol_Revision is present AND Protocol_Revision >= 13) THEN
 VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
4. MAKE (a condition exist that will cause the object to generate a fault condition)
5. BEFORE **Notification Fail Time**
 RECEIVE ConfirmedEventNotification-Request,
 'Process Identifier' = (PI3: any valid process ID),
 'Initiating Device Identifier' = IUT,
 'Event Object Identifier' = (the event-generating object configured for this test),
 'Time Stamp' = (Tfault: any valid time stamp),
 'Notification Class' = (the class corresponding to the object being tested),
 'Priority' = (Pfault: 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' = (the notify type configured for this event),
 'AckRequired' = TRUE,

```

    'From State' = NORMAL,
    'To State' = FAULT,
    'Event Values' = (values appropriate to the event type)
6. TRANSMIT BACnet-SimpleACK-PDU
7. VERIFY pCurrentState = FAULT
8. VERIFY Acked_Transitions = (?, FALSE, ?)
9. IF (Protocol_revision is present AND Protocol_Revision >= 13 THEN
    VERIFY Status_Flags = (FALSE, TRUE, ?, ?)
10. TRANSMIT AcknowledgeAlarm-Request,
    'Acknowledging Process Identifier' = (PI3),
    'Event Object Identifier' = (the event-generating object configured for this test),
    'Event State Acknowledged' = FAULT,
    'Acknowledgement Source' = (a character string),
    'Time Stamp' = (Tfault),
    'Time of Acknowledgment' = (the TD's current time)
11. RECEIVE BACnet-SimpleACK-PDU
12. IF (Protocol_Revision is present AND Protocol_Revision ≥ 1) THEN
    BEFORE Notification Fail Time
        RECEIVE ConfirmedEventNotification-Request,
            'Process Identifier' = (PI3),
            'Initiating Device Identifier' = IUT,
            'Event Object Identifier' = (the event-generating object configured for this test),
            'Time Stamp' = (Tfault the IUT's current time or sequence number),
            'Notification Class' = (the class corresponding to the object being tested),
            'Priority' = (Pfault),
            'Event Type' = IF (Protocol_Revision < 13)
                (any valid event type),
            ELSE
                CHANGE_OF_RELIABILITY,
            'Message Text' = (optional, any valid message text),
            'Notify Type' = ACK_NOTIFICATION,
            'To State' = FAULT
    ELSE
        BEFORE Notification Fail Time
            RECEIVE ConfirmedEventNotification-Request,
                'Process Identifier' = (PI3),
                'Initiating Device Identifier' = IUT,
                'Event Object Identifier' = (the event-generating object configured for this test),
                'Time Stamp' = (Tfault the IUT's current time or sequence number),
                'Notification Class' = (the class corresponding to the object being tested),
                'Priority' = (Pfault),
                'Event Type' = (any valid event type),
                'Message Text' = (optional, any valid message text),
                'Notify Type' = ACK_NOTIFICATION
13. TRANSMIT BACnet-SimpleACK-PDU
14. VERIFY Acked_Transitions = (?, TRUE, ?)

```

Problem:

After generating a fault condition, the status flags shall be observed in step 9. Usually the first flag (In-Alarm) will be 'TRUE' and not 'FALSE'. In similar tests (e.g. 7.3.1.1.1 Out_Of_Service, Status_Flags, and Reliability Test), the first flag is verified to be '?'.

Question:

Should step 9 be changed to:

9. IF (Protocol_revision is present AND Protocol_Revision >= 13 THEN
 VERIFY Status_Flags = (?, TRUE, ?, ?)

Response:

Yes.