



Testing and Conformance Clarification-Request No.: 20071119-37

(Request-Nr, assigned by moderator: yyyyymmxx where xx=sequence# within the month)

Request from: Frank.Schubert@mbs-software.de
(From BTF Intensive Workshop in Krefeld)

Stage:

- Request
- Listed by moderator
- Analysis by (TGTC or individual):.....
- Resolved

Reference: [referenced document(s) with number and revision]

7.3.1.11 requires a TimeDelay for TO_FAULT alarms

Background:

In step 20 the test requires a WAIT(TimeDelay), TO_FAULT alarms shall be reported immediately.

Question / proposed solution:

Change WAIT(TimeDelay) to read WAIT(InternalProcessingFailTime).

Response:

[By BIG-EU TGTC or by BTL-WG]

7.3.1.11 Acked_Transitions Tests

Reason for Change: This test has had the timestamp in the ack notifications changed to the current time of the initiating device. These test changes are covered by CN-060.

Dependencies: ConfirmedEventNotification Service Initiation Tests, 8.4; UnconfirmedEventNotification Service Initiation Tests, 8.5; AcknowledgeAlarm Service Execution Tests, 9.1; ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: 12.1.24, 12.2.25, 12.3.21, 12.5.23, 12.6.27, 12.7.25, 12.11.11, 12.14.19, 12.15.19, 12.16.34, 12.17.18, 12.18.19, 12.19.19 and 12.23.24.

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. This test applies to Event Enrollment objects and Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value, Life Safety Point, Life Safety Zone, Loop, Multi-state Input, Multi-state Output, and Multi-state Value objects that support intrinsic reporting.

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.

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. If a Notification Class object is being used to configure recipient information the value of the Transitions parameter for all recipients shall be (TRUE, TRUE, TRUE).

In the test description below, “X” is used to designate the event-triggering property.

Test Steps:

Test Steps:

1. WAIT (Time_Delay + **Notification Fail Time**)
2. VERIFY Event_State = NORMAL
3. VERIFY Acked_Transitions = (TRUE, TRUE, TRUE)
4. VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
5. IF (X is writable) THEN
 - WRITE X = (a value that is OFFNORMAL)
 - ELSE
 - MAKE (X have a value that is OFFNORMAL)
6. WAIT (Time_Delay)
7. BEFORE **Notification Fail Time**
 - 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,
 - 'Event Values' = (values appropriate to the event type)
8. VERIFY Event_State = OFFNORMAL
9. VERIFY Acked_Transitions = (FALSE, TRUE, TRUE)
10. VERIFY Status_Flags = (TRUE, FALSE, ?, ?)
11. IF (X is writable) THEN
 - WRITE X = (a value that is NORMAL)
 - ELSE
 - MAKE (X have a value that is NORMAL)
12. WAIT (Time_Delay)
13. BEFORE **Notification Fail Time**
 - 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),

- transition),
- 'Priority' = (the value configured to correspond to a TO-NORMAL)
- 'Event Type' = (any valid event type),
- 'Notify Type' = EVENT | ALARM,
- 'AckRequired' = TRUE | FALSE,
- 'From State' = OFNORMAL,
- 'To State' = NORMAL,
- 'Event Values' = (values appropriate to the event type)
14. VERIFY Event_State = NORMAL
 15. VERIFY Acked_Transitions = (FALSE, TRUE, FALSE)
 16. VERIFY Status_Flags = (FALSE, FALSE, ?,?)
 17. IF (the event-triggering object can be placed into a fault condition) THEN {
 18. MAKE (the event-triggering object change to a fault condition)
 19. ~~WAIT (Time_Delay)~~
 20. **BEFORE Notification Fail Time**
 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' = (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)
 21. VERIFY Event_State = FAULT
 22. VERIFY Acked_Transitions = (FALSE, FALSE, FALSE)
 23. VERIFY Status_Flags = (FALSE, TRUE, ?, ?)
 24. MAKE (the event-triggering object change to a normal condition)
 25. WAIT (Time_Delay)
 26. **BEFORE Notification Fail Time**
 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),
 'Notify Type' = EVENT | ALARM,
 'AckRequired' = TRUE | FALSE,
 'From State' = FAULT,
 'To State' = NORMAL,
 'Event Values' = (values appropriate to the event type)
 27. VERIFY Event_State = NORMAL
 28. VERIFY Acked_Transitions = (FALSE, FALSE, FALSE)
 29. VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
 30. TRANSMIT AcknowledgeAlarm-Request,
 'Acknowledging Process Identifier' = (the value of the 'Process Identifier' in step 20),
 'Event Object Identifier' = (the 'Event Object Identifier' in step 20),

```

    'Event State Acknowledged' =    FAULT,
    'Time Stamp' =                  (the 'Time Stamp' in step 20),
    'Time of Acknowledgment' =     (the current time)
31.  RECEIVE BACnet-SimpleACK-PDU
32.  IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
    RECEIVE ConfirmedEventNotification-Request,
    'Process Identifier' =          (the value of the 'Process Identifier' in step 20),
    'Initiating Device Identifier' = IUT,
    'Event Object Identifier' =     (the 'Event Object Identifier' in step 20),
    'Time Stamp' =                  (the current time or sequence number),
    'Notification Class' =         (the 'Notification Class' in step 20),
    'Priority' =                    (the 'Priority' in step 20),
    'Event Type' =                  (the 'Event Type' in step 20),
    'Notify Type' =                 ACK_NOTIFICATION,
    'To State' =                    FAULT
ELSE
    RECEIVE ConfirmedEventNotification-Request,
    'Process Identifier' =          (the value of the 'Process Identifier' in step 20),
    'Initiating Device Identifier' = IUT,
    'Event Object Identifier' =     (the 'Event Object Identifier' in step 20),
    'Time Stamp' =                  (the current time or sequence number) ,
    'Notification Class' =         (the 'Notification Class' in step 20),
    'Priority' =                    (the 'Priority' in step 20),
    'Event Type' =                  (the 'Event Type' in step 20),
    'Notify Type' =                 ACK_NOTIFICATION
33.  VERIFY Acked_Transitions = (FALSE, TRUE, FALSE)
34.  TRANSMIT AcknowledgeAlarm-Request,
    'Acknowledging Process Identifier' = (the value of the 'Process Identifier' in step 26),
    'Event Object Identifier' =         (the 'Event Object Identifier' in step 26),
    'Event State Acknowledged' =       NORMAL,
    'Time Stamp' =                     (the 'Time Stamp' in step 26),
    'Time of Acknowledgment' =         (the current time)
35.  RECEIVE BACnet-SimpleACK-PDU
36.  IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
    BEFORE Notification Fail Time
    RECEIVE ConfirmedEventNotification-Request,
    'Process Identifier' =          (the value of the 'Process Identifier' in step 26),
    'Initiating Device Identifier' = IUT,
    'Event Object Identifier' =     (the 'Event Object Identifier' in step 26),
    'Time Stamp' =                  (the current time or sequence number),
    'Notification Class' =         (the 'Notification Class' in step 26),
    'Priority' =                    (the 'Priority' in step 26),
    'Event Type' =                  (the 'Event Type' in step 26),
    'Notify Type' =                 ACK_NOTIFICATION,
    'To State' =                    NORMAL
ELSE
    BEFORE Notification Fail Time
    RECEIVE ConfirmedEventNotification-Request,
    'Process Identifier' =          (the value of the 'Process Identifier' in step 26),
    'Initiating Device Identifier' = IUT,
    'Event Object Identifier' =     (the 'Event Object Identifier' in step 26),
    'Time Stamp' =                  (the current time or sequence number),
    'Notification Class' =         (the 'Notification Class' in step 26),
    'Priority' =                    (the 'Priority' in step 26),
    'Event Type' =                  (the 'Event Type' in step 26),

```

```

        'Notify Type' =                ACK_NOTIFICATION
37. VERIFY Acked_Transitions = (FALSE, TRUE, TRUE)
38. TRANSMIT AcknowledgeAlarm-Request,
    'Acknowledging Process Identifier' = (the value of the 'Process Identifier' in step 7),
    'Event Object Identifier' =         (the 'Event Object Identifier' in step 7),
    'Event State Acknowledged' =       OFFNORMAL,
    'Time Stamp' =                     (the 'Time Stamp' in step 7),
    'Time of Acknowledgment' =         (the current time)
39. RECEIVE BACnet-SimpleACK-PDU
40. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
    BEFORE Notification Fail Time
    RECEIVE ConfirmedEventNotification-Request,
        'Process Identifier' =         (the value of the 'Process Identifier' in step 7),
        'Initiating Device Identifier' = IUT,
        'Event Object Identifier' =     (the 'Event Object Identifier' in step 7),
        'Time Stamp' =                 (the current time or sequence number),
        'Notification Class' =         (the 'Notification Class' in step 7),
        'Priority' =                   (the 'Priority' in step 7),
        'Event Type' =                 (the 'Event Type' in step 7),
        'Notify Type' =                ACK_NOTIFICATION,
        'To State' =                   OFFNORMAL
    ELSE
    BEFORE Notification Fail Time
    RECEIVE ConfirmedEventNotification-Request,
        'Process Identifier' =         (the value of the 'Process Identifier' in step 7),
        'Initiating Device Identifier' = IUT,
        'Event Object Identifier' =     (the 'Event Object Identifier' in step 7),
        'Time Stamp' =                 (the current time or sequence number),
        'Notification Class' =         (the 'Notification Class' in step 7),
        'Priority' =                   (the 'Priority' in step 7),
        'Event Type' =                 (the 'Event Type' in step 7),
        'Notify Type' =                ACK_NOTIFICATION
41. VERIFY Acked_Transitions = (TRUE, TRUE, TRUE)

```

Notes to Tester: The UnconfirmedEventNotification service may be substituted for the ConfirmedEventNotification service. The 'Message Text' parameter is omitted in the test description because it is optional. The IUT may include this parameter in the notification messages.