

Clarification Request

References: ASHRAE Standard 135-2012 (Section 12.4.7); BTL Specified Tests-12.0.final, 7.3.1.10 Event_Enable Tests

Date of BTL-WG Response: August 21, 2014

☒ All actions necessitated have been completed

Background:

There appears to be a conflict between the Standard and BTL Test 7.3.1.10. Although it is quite common for the event-state to track a potential event, the standard clearly indicates that it should only track active events. A device, configured with an event-enable property value = (0,0,1) is disabling TO-OFFNORMAL and TO_FAULT events. Therefore, it seems logical that OFFNORMAL values would not change the event-state property because there exists no 'active' event. Steps 6, 10, and 11 in BTL Test 7.3.1.10 dictate that the event-state property track potential events and I think this is incorrect.

From 135-2012, Section 12.4.7:

The Event_State property, of type BACnetEventState, is included in order to provide a way to determine whether this object has an active event state associated with it (see Clause 13.2.2.1). If the object supports event reporting, then the Event_State property shall indicate the event state of the object. If the object does not support event reporting then the value of this property shall be NORMAL.

BTL test 7.3.1.10 – Event_Enable Tests

Reason For Change: The test does not call out what to do if the Event_Enable property is read-only and the IUT cannot be configured as specified in the Configuration Requirements. This test also contains an error in step 11. There should not be a wait after the event-triggering property is put into a FAULT state. There is no SSPC proposal for this test.

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

BACnet Reference Clauses: 12.1.23, 12.2.24, 12.3.20, 12.5.22, 12.6.26, 12.7.24, 12.11.10, 12.14.18, 12.15.18, 12.16.33, 12.17.17, 12.18.18, 12.19.18 and 12.23.23.

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 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 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 Event_Enable property has a value of TRUE.

Configuration Requirements: The Event_Enable property 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. If the Event_Enable property is not configurable, follow the test steps as written and verify correct behavior for the value of the Event_Enable property. 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.

1. VERIFY Event_State = NORMAL
2. WAIT (Time_Delay + **Notification Fail Time**)
3. IF (X is writable) THEN
 - WRITE X = (a value that is OFFNORMAL)
 - ELSE
 - MAKE (X have a value that is OFFNORMAL)
4. WAIT (Time_Delay)
5. BEFORE **Notification Fail Time**
 - IF (the Transitions bit corresponding to the 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' = (the current local time),
 - '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)
 - ELSE
 - CHECK (verify that the IUT did not transmit an event notification message)
 - 6. VERIFY Event_State = OFFNORMAL
 - 7. IF (X is writable) THEN
 - WRITE X = (a value that is NORMAL)
 - ELSE
 - MAKE (X have a value that is NORMAL)
 - 8. WAIT (Time_Delay)
 - 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' = (the current local time),
 - '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' = OFFNORMAL,
 - 'To State' = NORMAL,
 - 'Event Values' = (values appropriate to the event type)
 - ELSE
 - CHECK (verify that the IUT did not transmit an event notification message)

10. VERIFY Event_State = NORMAL

11. IF (the event-triggering object can be placed into a fault condition) THEN {

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

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' = (the current local time),

'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)

ELSE

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

VERIFY Event_State = FAULT

}

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.

Question:

Should the 'VERIFY Event_State = XX' in Steps 6, 10, and 11 be moved inside the IF-THEN statements so they are only checked if the event-state would correspond to an active event?

Response:

"No, the test is correct. Clause 13.2.2 which describes detection makes no reference to the Event_Enable property, while clause 13.2.5 regarding distribution does reference the Event_Enable property. The three flags in the Event_Enable property separately enable and disable the distribution. The Event_State should still change."