Clarification Request

Request from: BTL Manager

Reference: "BTL Specified Tests-3.0.final", "ASHRAE 135.1-2003"

Background:

The ASHRAE 135.1 – 8.4.2 is written to verify the operation of the CHANGE_OF_STATE algorithm. The test applies to Event Enrollment Objects as well as for intrinsically reporting Binary Input, Binary Value, Multi-state Input and Multi-state Value objects.

This test is confusing when trying to run the test using an Event Enrollment Object. Step 6 is a VERIFY of the Status_Flags property. The Event Enrollment object does not contain a Status_Flags property. If the reference object for the Event Enrollment object does not support intrinsic reporting, the Status_Flags property is not required to indicate 'in_alarm'. When this test is run on an Event Enrollment object, should this step be ignored if the referenced object does not support intrinsic alarming?

Here is the complete test for your reference:

Purpose: This test case verifies the correct operation of the CHANGE_OF_STATE event algorithm. It applies to Event Enrollment objects with an Event_Type of CHANGE_OF_STATE and to intrinsic event reporting for Binary Input, Binary Value, Multi-state Input and Multi-state Value objects.

Test Concept: The object begins the test in a NORMAL state. The Present_Value (referenced property) is changed to a value that is one of the values designated in List_Of_Values. After the time delay expires the object should enter the OFFNORMAL state and transmit an event notification message. The Present_Value (referenced property) is then changed to a value corresponding to a NORMAL state. After the time delay the object should enter the NORMAL state and transmit an event notification message. For Multi-state Input and Multi-state Value objects there is a special case of the CHANGE_OF_STATE algorithm that applies to transitions to the FAULT state. The test procedure includes a test for this special case.

Configuration Requirements: The IUT shall be configured such that the Event_Enable property has a value of TRUE for the TO-OFFNORMAL, TO-FAULT and TO-NORMAL transitions. The Issue_Confirmed_Notifications property shall have a value of TRUE. The event-generating objects shall be in a NORMAL state at the start of the test.

In the test description below Present_Value is used as the referenced property. If an Event Enrollment object is being tested Present_Value should be replaced by the appropriate property reference.

Test Steps:
1. VERIFY Event_State = NORMAL
2. IF (Present_Value is writable) THEN
    WRITE Present_Value = (a value x such that x = Alarm_Value for binary objects or one of the Alarm_Values for multi-state objects)

ELSE
    MAKE (Present_Value have a value x such that x = Alarm_Value for binary objects or one of the Alarm_Values for multi-state objects)
3. WAIT Time_Delay
4. BEFORE **Notification Fail Time**
    RECEIVE ConfirmedEventNotification-Request,
        'Process Identifier' = (any valid process ID),
        'Initiating Device Identifier' = IUT,

'Event Object Identifier' = (the intrinsic reporting object being tested or the Event Enrollment object being tested),
'Time Stamp' = (the current local time),
'Notification Class' = (the configured notification class),
'Priority' = (the value configured to correspond to a TO-OFFNORMAL transition),
'Event Type' = CHANGE_OF_STATE,
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = OFFNORMAL,
' Event Values' = Present_Value, Status_Flags

5. TRANSMIT BACnet-SimpleACK-PDU
6. VERIFY Status_Flags = (TRUE, FALSE, FALSE, FALSE)
7. VERIFY Event_State = OFFNORMAL
8. IF (Protocol_Revision is present and Protocol_Revision >= 1) THEN
   VERIFY Event_Time_Stamps = (the timestamp in step 4, *, *)
9. IF (Present_Value is writable) THEN
   WRITE Present_Value = (a value x such that x corresponds to a NORMAL state)
   ELSE
   MAKE (Present_Value have a value x such that x corresponds to a NORMAL state)
10. WAIT Time_Delay
11. BEFORE **Notification Fail Time**
   RECEIVE ConfirmedEventNotification-Request,
   'Process Identifier' = (any valid process ID),
   'Initiating Device Identifier' = IUT,
   'Event Object Identifier' = (the intrinsic reporting object being tested or the object referenced by the Event Enrollment object being tested),
   'Time Stamp' = (the current local time),
   'Notification Class' = (the configured notification class),
   'Priority' = (the value configured to correspond to a TO-NORMAL transition),
   'Event Type' = CHANGE_OF_STATE,
   'Notify Type' = EVENT | ALARM,
   'AckRequired' = TRUE | FALSE,
   'From State' = OFFNORMAL,
   'To State' = NORMAL,
   ' Event Values' = Present_Value, Status_Flags
12. TRANSMIT BACnet-SimpleACK-PDU
13. VERIFY Status_Flags = (FALSE, FALSE, FALSE, FALSE)
14. VERIFY Event_State = NORMAL
15. IF (Protocol_Revision is present and Protocol_Revision >= 1) THEN
   VERIFY Event_Time_Stamps = (the timestamp in step 4, *, the timestamp in step 11)
16. IF (the object being tested is a multi-state object that supports intrinsic reporting) THEN
17. IF (Present_Value is writable) THEN
   WRITE Present_Value = (a value x such that x = one of the Fault_Values)
   ELSE
   MAKE (Present_Value have a value x such that x = one of the Fault_Values)
18. WAIT Time_Delay
19. BEFORE **Notification Fail Time**
   RECEIVE ConfirmedEventNotification-Request,
   'Process Identifier' = (any valid process ID),
   'Initiating Device Identifier' = IUT,
   'Event Object Identifier' = (the intrinsic reporting object being tested),
   'Time Stamp' = (the current local time),
   'Notification Class' = (the configured notification class),
   'Priority' = (the value configured to correspond to a TO-FAULT transition),
   'Event Type' = CHANGE_OF_STATE,

'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = FAULT,
'Event Values' = Present_Value, Status_Flags

20. TRANSMIT BACnet-SimpleACK-PDU
21. VERIFY Status_Flags = (~~FALSE~~*TRUE*, TRUE, FALSE, FALSE)
22. VERIFY Event_State = FAULT
23. IF (Protocol_Revision is present and Protocol_Revision >= 1) THEN
VERIFY Event_Time_Stamps = (the timestamp in step 4, the timestamp in step 19, the timestamp in step 11)
24. IF (Present_Value is writable) THEN
WRITE Present_Value = (a value x such that x corresponds to a NORMAL state)
ELSE
MAKE (Present_Value have a value x such that x corresponds to a NORMAL state)
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 intrinsic reporting object being tested),
'Time Stamp' = (the current local time),
'Notification Class' = (the configured notification class),
'Priority' = (the value configured to correspond to a TO-NORMAL transition),
'Event Type' = CHANGE_OF_STATE,
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = FAULT,
'To State' = NORMAL,
' Event Values' = Present_Value, Status_Flags
27. TRANSMIT BACnet-SimpleACK-PDU
28. VERIFY Status_Flags = (FALSE, FALSE, FALSE, FALSE)
29. VERIFY Event_State = NORMAL
30. IF (Protocol_Revision is present and Protocol_Revision >= 1) THEN
31. VERIFY Event_Time_Stamps = (the timestamp in step 4, the timestamp in step 19, the timestamp in step 26)

Passing Result: The 'Message Text' parameter is omitted in the test description because it is optional. The IUT may include this parameter in the notification messages. The time stamps indicated by "*" in steps 8 and 15 can have a value that indicates an unspecified time or a time that precedes the timestamp in step 4.

Response:

The BTL working group agrees that if the object being tested is an Event Enrollment object, the steps that verify the Status_Flags property shall be skipped. The BTL working group will create a new test with this correction and submit the changes to the SSPC.