Clarification Request

References: 135.1.2013 -8.4.4 and 135.1-2013 -8.5.4 in Test Plan section 5.2.11 and 5.3.9

Date of BTL-WG Response: __09-Aug-2018__

☒All Actions Necessitated have been Completed

Background: 135.1.2013 - 8.4.4 and 8.5.4 COMMAND_FAILURE

1. VERIFY Event_State = NORMAL
2. IF (the object being tested is not an Event Enrollment object) THEN
VERIFY Status_Flags = ~~(FALSE, FALSE, FALSE, FALSE)~~ (FALSE, FALSE, ?, ?)
3. IF (Present_Value is writable) THEN
WRITE Present_Value = (a different value)
ELSE
MAKE (Present_Value take on a different value)
4. WAIT (Time_Delay)
5. BEFORE **Notification Fail Time**
RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the 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' = COMMAND_FAILURE,
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = OFFNORMAL,
'Event Values' = Present_Value, Status_Flags, Feedback_Value
6. TRANSMIT BACnet-SimpleACK-PDU
7. IF (the object being tested is not an Event Enrollment object) THEN
VERIFY Status_Flags = (TRUE, FALSE, ?, ?)
8. VERIFY Event_State = OFFNORMAL
9. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
VERIFY Event_Time_Stamps = (the timestamp in step 5, *, *)
10. IF (Feedback_Value is writable) THEN
WRITE Feedback_Value = (a value consistent with Present_Value)
ELSE
MAKE (Feedback_Value take on a value consistent with Present_Value)
11. WAIT (Time_Delay)
12. BEFORE **Notification Fail Time**
RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the 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' = COMMAND_FAILURE,
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = OFFNORMAL,
'To State' = NORMAL,
'Event Values' = Present_Value, Status_Flags, Feedback_Value

13. TRANSMIT BACnet-SimpleACK-PDU
14. IF (the object being tested is not an Event Enrollment object) THEN
VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
15. VERIFY Event_State = NORMAL
16. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
VERIFY Event_Time_Stamps = (the timestamp in step 5, *, the timestamp in step 12)

Problem:

The test fails in Step 2 for a device that sets the overridden Flag to ‘TRUE’ when decoupling the feedback_value from the physical input.
It isn't clear from the standard, why the restriction on the Status_Flags are so sharp here and ‘(FALSE, FALSE, FALSE, FALSE)’ is required. Later in the same test, the restrictions are '(TRUE, FALSE, ?, ?)' and '(FALSE, FALSE, ?, ?)'.

Question:

Is the test too strict in Step 2 expecting Status_Flags to be ‘(FALSE, FALSE, FALSE, FALSE)’?

Should step 2 of the test be changed to
2. IF (the object being tested is not an Event Enrollment object) THEN
VERIFY Status_Flags = (FALSE, FALSE, ?, ?)

Response:

YES.

YES