Clarification Request

Request from: BTL Manager (btl-manager@bacnetinternational.org)

Reference: "ASHRAE 135.1-2003", "ASHRAE 135-2004"

Background:

Test 135.1 - 8.4.3.1 and 8.5.3.1 Numerical Algorithm are written specifically for EventEnrollment objects. The BTL Test Plan uses this test to apply toward any event initiating object. The problem I see with this is that the test specifically uses the property 'referenced-property-increment' which only exists in the EE object. According the 135-2004, Table 13-1 pg 254, Analog objects trigger a CHANGE_OF_VALUE event when the Present_Value changes by COV_Increment or if the Status_Flags change at all.

We need to either:

1) Change the test to conditionally use the EE object properties or the Analog object properties
2) OR Change the BTL Test Plan document to state clearly in the Test Conditionality section what parts of the test should be changed in order to perform this specific test on an Analog Object.
3) OR Add a new test that uses Analog objects and change our test plan to conditionally use the appropriate tests based on the objects that are contained in the IUT.

I would recommend the first option in order to make sure that the test in 135.1 is accurate and we don't have to rely on the test conditionality of the Test Plan document.

Proposed Solution:

8.4.3.1 Numerical Algorithm

The test in this clause applies to use of the CHANGE_OF_VALUE algorithm applied to ~~Integer or~~ Real datatypes.

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: 12.11, 13.3.2, and 13.8.

Purpose: To verify the correct operation of the CHANGE_OF_VALUE event algorithm as applied to numerical datatypes. This test applies to ~~Event-Enrollment objects~~ *any event initiating object* with an Event_Type of CHANGE_OF_VALUE.

Test Concept: The object begins the test in a NORMAL state. The referenced property is changed by a value that is less than the Referenced_Property_Increment *(or COV_Increment for Analog Objects)*. The tester verifies that no event notification is transmitted. The referenced property is changed again to a value that differs from the original value by an amount greater than the Referenced_Property_Increment *(or COV_Increment for Analog Objects)*. The tester verifies that an event notification message is transmitted and that the proper Event_State transitions occur.

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

Test Steps:

1. VERIFY Event_State = NORMAL
2. IF (the referenced property is writable) THEN
   WRITE (referenced property) = (a value x: x differs from the initial value by less than Referenced_Property_Increment *(or COV_Increment for Analog Objects)*)
   ELSE
   MAKE (the referenced property have a value x: x differs from the initial value by less than Referenced_Property_Increment *(or COV_Increment for Analog Objects)*)
3. WAIT (Time_Delay + **Notification Fail Time**)
4. CHECK (verify that no event notification message is transmitted)
5. IF (the referenced property is writable) THEN
   WRITE (referenced property) = (a value x: x differs from the initial value in step 1 by more than Referenced_Property_Increment *(or COV_Increment for Analog Objects)*)
   ELSE
   MAKE (the referenced property have a value x: x differs from the initial value in step 1 by more than Referenced_Property_Increment *(or COV_Increment for Analog Objects)*)
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 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_VALUE,
   'Notify Type' = EVENT | ALARM,
   'AckRequired' = TRUE | FALSE,
   'From State' = NORMAL,
   'To State' = NORMAL,
   ' Event Values' = changed-value, Status_Flags
8. TRANSMIT BACnet-SimpleACK-PDU
9. VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
10. VERIFY Event_State = NORMAL
11. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
    VERIFY Event_Time_Stamps = (*, *, the timestamp in step 7)

Notes to Tester: 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 step 11 can have a value that indicates an unspecified time or a time that precedes the timestamp in step 7.

Question:

Is the proposed solution acceptable?

Response:

No. The CHANGE_OF_VALUE event can only be generated from an Event Enrollment object or a Proprietary object. Table 13-2 in the standard lists the algorithms that are implemented by each object type if it performs intrinsic alarming and should not be confused with the COV Notifications which use a different service.
Please note that the reference to Integer data types should be removed from the test description.