

Clarification Request

References: BTL Specified Test 8.4.X4

Date of BTL-WG Response: April 18, 2013

Background:

8.4.X4 CHANGE_OF_CHARACTERSTRING Tests (ConfirmedEventNotification)

Reason for Change: New algorithm for Protocol_Revision 10.

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

BACnet Reference Clauses: <update these as appropriate>

Purpose: To verify the correct operation of the CHANGE_OF_CHARACTERSTRING event algorithm. This test applies to Event Enrollment objects with an Event_Type of CHANGE_OF_CHARACTERSTRING and to intrinsic event reporting for CharacterString 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 different value in the 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. The transition to and from FAULT is also tested.

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.

The object shall be configured with a non-empty Alarm_Values property and a non-empty Fault_Values property if possible.

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 (the object, or referenced object, if using Event Enrollment, has a non-empty Alarm_Values property) THEN
3. IF (Present_Value is writable) THEN
 - WRITE Present_Value = (a value x: x = one of the Alarm_Values)
- ELSE
 - MAKE (Present_Value have a value x: x = one of the Alarm_Values)
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 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_CHARACTERSTRING,
 'Notify Type' = EVENT | ALARM,
 'AckRequired' = TRUE | FALSE,
 'From State' = NORMAL,
 'To State' = OFFNORMAL,
 'Event Values' = Present_Value, Status_Flags
 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. VERIFY Event_Time_Stamps = (the timestamp in step 5), ARRAYINDEX = 1
 10. IF (the object, or referenced object, if using Event Enrollment, has a Alarm_Values property with more than 1 entry) THEN
 11. IF (Present_Value is writable) THEN
 WRITE Present_Value = (a value x: x = one of the Alarm_Values not used in prior steps)
 ELSE
 MAKE (Present_Value have a value x: x = one of the Alarm_Values not used in prior steps)
 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 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_CHARACTERSTRING,
 'Notify Type' = EVENT | ALARM,
 'AckRequired' = TRUE | FALSE,
 'From State' = OFFNORMAL,
 'To State' = OFFNORMAL,
 'Event Values' = Present_Value, Status_Flags
 14. TRANSMIT BACnet-SimpleACK-PDU
 15. IF (the object being tested is NOT an Event Enrollment object) THEN
 VERIFY Status_Flags = (TRUE, FALSE,?,?)
 16. VERIFY Event_State = OFFNORMAL
 17. VERIFY Event_Time_Stamps = (the timestamp in step 13), ARRAYINDEX = 1
 18. IF (the object, or referenced object, if using Event Enrollment, has a non-empty Alarm_Values property) THEN
 18. IF (Present_Value is writable) THEN
 WRITE Present_Value = (a value x: x corresponds to a NORMAL state)
 ELSE
 MAKE (Present_Value have a value x: x corresponds to a NORMAL state)
 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 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_CHARACTERSTRING,
 'Notify Type' = EVENT | ALARM,
 'AckRequired' = TRUE | FALSE,
 'From State' = OFFNORMAL,
 'To State' = NORMAL,
 'Event Values' = Present_Value, Status_Flags
 21. TRANSMIT BACnet-SimpleACK-PDU
 22. IF (the object being tested is NOT an Event Enrollment object) THEN
 VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
 23. VERIFY Event_State = NORMAL
 24. VERIFY Event_Time_Stamps = (the timestamp in step 20), ARRAYINDEX = 3
 25. IF (the object, or referenced object, if testing Event Enrollment, is configured with a non-empty Fault_Values property) THEN
 26. IF (Present_Value is writable) THEN
 WRITE Present_Value = (a value x: x = one of the Fault_Values)
 ELSE
 MAKE (Present_Value have a value x: x = one of the Fault_Values)
 27. WAIT (Time_Delay)
 28. 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_CHARACTERSTRING,
 'Notify Type' = EVENT | ALARM,
 'AckRequired' = TRUE | FALSE,
 'From State' = NORMAL,
 'To State' = FAULT,
 'Event Values' = Present_Value, Status_Flags
 29. TRANSMIT BACnet-SimpleACK-PDU
 30. IF (the object being tested is NOT an Event Enrollment object) THEN
 VERIFY Status_Flags = (TRUE, TRUE, ?, ?)
 31. VERIFY Event_State = FAULT
 32. VERIFY Event_Time_Stamps = (the timestamp in step 13, the timestamp in step 28, the timestamp in step 20)
 33. VERIFY Reliability = MULTI_STATE_FAULT
 34. IF (Present_Value is writable) THEN
 WRITE Present_Value = (a value x: x corresponds to a NORMAL state)
 ELSE
 MAKE (Present_Value have a value x: x corresponds to a NORMAL state)
 35. WAIT (Time_Delay)
 36. 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_CHARACTERSTRING,
 'Notify Type' = EVENT | ALARM,
 'AckRequired' = TRUE | FALSE,
 'From State' = FAULT,
 'To State' = NORMAL,
 'Event Values' = Present_Value, Status_Flags
 37. TRANSMIT BACnet-SimpleACK-PDU
 38. IF (the object being tested is NOT an Event Enrollment object) THEN
 VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
 39. VERIFY Event_State = NORMAL
 40. VERIFY Event_Time_Stamps = (the timestamp in step 28), ARRAYINDEX = 2

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 steps 9 and 17 can have a value that indicates an unspecified time or a time that precedes the timestamp in step 5.

Problems:

In step 5 a Normal To Offnormal Transition is expected.

In step 13 another Normal To Offnormal Transition is expected without any To Normal in between (problem seems to start with the test concept).

Step 25

IF (the object, or referenced object, if testing Event Enrollment, is configured with a non-empty Fault_Values property) THEN

seems to suggest, that there might be Fault_Values that apply to EventEnrollment. Especially the phrase "or referenced object, if testing Event Enrollment" looks very strange. But an Event Enrollment with "change-of-characterstring" does not have access to any FaultValues!

Step 30

VERIFY Status_Flags = (TRUE, TRUE, ?, ?)

expects the Alarm Bit be have a value of TRUE.

But the object has just gone from normal to fault. So the AlarmBit in StatusFlags should probably be FALSE!

Question:

How should the test be performed?

Response:

Issue #1: In step 5, the transition expected should be OFFNORMAL to OFFNORMAL.

Issue #2: This is not a problem as the Fault_Values property in the referenced object will control the value of the Reliability of the referenced object and thus will result in the Event Enrollment object transitioning to FAULT.

Issue #3: The test is correct as written. The InAlarm bit is set whenever the Event_State property is not NORMAL so it will be set when the Event_State is FAULT.

The BTL-WEG noted another issue with the test and the corrected test steps are shown below:

8.4.X4 CHANGE_OF_CHARACTERSTRING Tests (ConfirmedEventNotification)

Reason for Change: New algorithm for Protocol_Revision 10.

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

BACnet Reference Clauses: <update these as appropriate>

Purpose: To verify the correct operation of the CHANGE_OF_CHARACTERSTRING event algorithm. This test applies to Event Enrollment objects with an Event_Type of CHANGE_OF_CHARACTERSTRING and to intrinsic event reporting for CharacterString 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 different value in the 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. The transition to and from FAULT is also tested.

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.

The object shall be configured with a non-empty Alarm_Values property and a non-empty Fault_Values property if possible.

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 (the object, or referenced object, if using Event Enrollment, has a non-empty Alarm_Values property) THEN

3. IF (Present_Value is writable) THEN
 WRITE Present_Value = (a value x: x = one of the Alarm_Values)
 ELSE
 MAKE (Present_Value have a value x: x = one of the Alarm_Values)
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 intrinsic reporting object being tested or the Event Enrollment object being tested),
 'Time Stamp' = (Toffnormal; 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_CHARACTERSTRING,
 'Notify Type' = EVENT | ALARM,
 'AckRequired' = TRUE | FALSE,
 'From State' = NORMAL,
 'To State' = OFFNORMAL,
 'Event Values' = Present_Value, Status_Flags
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. VERIFY Event_Time_Stamps = (Toffnormal, *, *)
10. IF (the object, or referenced object, if using Event Enrollment, has a Alarm_Values property with more than 1 entry) THEN
11. IF (Present_Value is writable) THEN
 WRITE Present_Value = (a value x: x = one of the Alarm_Values not used in prior steps)
 ELSE
 MAKE (Present_Value have a value x: x = one of the Alarm_Values not used in prior steps)
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 intrinsic reporting object being tested or the Event Enrollment object being tested),
 'Time Stamp' = (Toffnormal; 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_CHARACTERSTRING,
 'Notify Type' = EVENT | ALARM,
 'AckRequired' = TRUE | FALSE,
 'From State' = OFFNORMAL,
 'To State' = OFFNORMAL,
 'Event Values' = Present_Value, Status_Flags
14. TRANSMIT BACnet-SimpleACK-PDU
15. IF (the object being tested is NOT an Event Enrollment object) THEN
 VERIFY Status_Flags = (TRUE, FALSE,?,?)
16. VERIFY Event_State = OFFNORMAL
17. VERIFY Event_Time_Stamps = (Toffnormal, *, *)

18. IF (the object, or referenced object, if using Event Enrollment, has a non-empty Alarm_Values property) THEN

19. IF (Present_Value is writable) THEN
 - WRITE Present_Value = (a value x: x corresponds to a NORMAL state)
 - ELSE
 - MAKE (Present_Value have a value x: x corresponds to a NORMAL state)
20. WAIT (Time_Delay)
21. 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' = (Tnormal: 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_CHARACTERSTRING,
 - 'Notify Type' = EVENT | ALARM,
 - 'AckRequired' = TRUE | FALSE,
 - 'From State' = OFFNORMAL,
 - 'To State' = NORMAL,
 - 'Event Values' = Present_Value, Status_Flags
22. TRANSMIT BACnet-SimpleACK-PDU
23. IF (the object being tested is NOT an Event Enrollment object) THEN
 - VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
24. VERIFY Event_State = NORMAL
25. VERIFY Event_Time_Stamps = (Toffnormal, *, Tnormal)
26. IF (the object, or referenced object, if testing Event Enrollment, is configured with a non-empty Fault_Values property) THEN
27. IF (Present_Value is writable) THEN
 - WRITE Present_Value = (a value x: x = one of the Fault_Values)
 - ELSE
 - MAKE (Present_Value have a value x: x = one of the Fault_Values)
28. WAIT (Time_Delay)
29. 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' = (Tfault: 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_CHARACTERSTRING,
 - 'Notify Type' = EVENT | ALARM,
 - 'AckRequired' = TRUE | FALSE,
 - 'From State' = NORMAL,
 - 'To State' = FAULT,
 - 'Event Values' = Present_Value, Status_Flags
30. TRANSMIT BACnet-SimpleACK-PDU
31. IF (the object being tested is NOT an Event Enrollment object) THEN
 - VERIFY Status_Flags = (TRUE, TRUE, ?, ?)
32. VERIFY Event_State = FAULT
33. VERIFY Event_Time_Stamps = (Toffnormal, Tfault, Tnormal)
34. VERIFY Reliability = MULTI_STATE_FAULT

35. IF (Present_Value is writable) THEN
 WRITE Present_Value = (a value x: x corresponds to a NORMAL state)
 ELSE
 MAKE (Present_Value have a value x: x corresponds to a NORMAL state)
36. WAIT (Time_Delay)
37. 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' = (Tfault: 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_CHARACTERSTRING,
 'Notify Type' = EVENT | ALARM,
 'AckRequired' = TRUE | FALSE,
 'From State' = FAULT,
 'To State' = NORMAL,
 'Event Values' = Present_Value, Status_Flags
38. TRANSMIT BACnet-SimpleACK-PDU
39. IF (the object being tested is NOT an Event Enrollment object) THEN
 VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
40. VERIFY Event_State = NORMAL
41. VERIFY Event_Time_Stamps = (Toffnormal, Tfault, Tnormal)

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 "*" can have a value that indicates an unspecified time or a time that precedes the timestamp of the first received notification.