

Clarification Request

References: Testsource: 135.1-2013

Date of BTL-WG Response: 04-Aug-2016

☒ All actions necessitated have been completed

Background:

In test 8.4.2 (ANSI/ASHRAE Standard 135.1-2013) step 2 there is an IF, that will not be closed for the whole test. In step 17 there is an IF indented so that it is part of the IF from step 2.

135.1-2013 version of the 8.4.2 CHANGE_OF_STATE Tests

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

BACnet Reference Clauses: 12.6, 12.8, 12.18, 12.20, 13.2, 13.3.2, and 13.8.

Purpose: To verify the correct operation of the CHANGE_OF_STATE event algorithm. This test 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 (the object, or referenced object, if using Event Enrollment, is a binary object or it is a multi-state object with
a non-empty Alarm_Values property) THEN
3. IF (Present_Value is writable) THEN

WRITE Present_Value = (a value x: x = Alarm_Value for binary objects or one of the Alarm_Values for

multi-state objects)

ELSE

MAKE (Present_Value have a value x: x = Alarm_Value for binary objects or one of the Alarm_Values

for multi-state objects)

4. WAIT (Time_Delay)

5. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,

'Process Identifier' = (any valid process ID),

'Initiating Device Identifier' = IUT,

Enrollment 'Event Object Identifier' = (the intrinsic reporting object being tested or the Event object being tested),

'Time Stamp' = (T1, 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

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 \geq 1) THEN

VERIFY Event_Time_Stamps = (T1, *, *)

10. 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)

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 intrinsic reporting object being tested or the Event
 Enrollment object being tested),
 'Time Stamp' = (T2, 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

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 \geq 1) THEN
 VERIFY Event_Time_Stamps = (T1, *, T2)
17. IF (the object being tested is a multi-state object that supports intrinsic reporting) THEN
18. 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)
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' = (T3, 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.    IF (the object being tested is NOT an Event Enrollment object) THEN
        VERIFY Status_Flags = (FALSE, TRUE, ?, ?)
22.    VERIFY Event_State = FAULT
23.    IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
        VERIFY Event_Time_Stamps = (T1, T3, T2)
24.    IF (the object being tested is a multi-state object that supports intrinsic reporting and
        Protocol_Revision is
            present and Protocol_Revision ≥ 1) THEN
        VERIFY Reliability = MULTI_STATE_FAULT
25.    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)
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' = (T4, 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.    IF (the object being tested is NOT an Event Enrollment object) THEN
        VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
29.    VERIFY Event_State = NORMAL
30.    IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
        VERIFY Event_Time_Stamps = (T1, T3, T4)

```

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 can have a value that indicates an unspecified time or a time that precedes the timestamp T1.

Probably both IFs should have the same indentation as was the case in the BTL Specified Tests-12.0.final version of the tests.

BTL Specified Tests-12.0.final version of the 8.4.2 CHANGE_OF_STATE Tests

Reason for Change: This test was incorrect when used to test an Event Enrollment Object. The change is in LJT-001.

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

BACnet Reference Clauses: 12.5, 12.7, 12.17, 12.19, 13.2,13.3.2, and 13.8.

Purpose: To verify the correct operation of the CHANGE_OF_STATE event algorithm. This test 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.

For multi-state objects, 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, is a binary object or a multi-state object with a non-empty Alarm_Values property) THEN
3. IF (Present_Value is writable) THEN
 - WRITE Present_Value = (a value x: x = Alarm_Value for binary objects or one of the Alarm_Values for multi-state objects)
- ELSE
 - MAKE (Present_Value have a value x: x = Alarm_Value for binary objects or one of the Alarm_Values for multi-state objects)
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_STATE,
 - '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. IF (Protocol_Revision is present and Protocol_Revision \geq 1) THEN
 - VERIFY Event_Time_Stamps = (the timestamp in step 5, *, *)
10. 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)
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 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
 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 \geq 1) THEN
 VERIFY Event_Time_Stamps = (the timestamp in step 5, *, the timestamp in
 step 12)
 17. IF (the object, or referenced object, if testing Event Enrollment, being tested is a
 multi-state object and is configured with a non-empty Fault_Values property) THEN
 18. 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)
 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),
 '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
 21. TRANSMIT BACnet-SimpleACK-PDU
 22. IF (the object being tested is NOT an Event Enrollment object) THEN
 VERIFY Status_Flags = (TRUE, TRUE, ?, ?)
 23. VERIFY Event_State = FAULT
 24. IF (Protocol_Revision is present and Protocol_Revision \geq 1) THEN
 VERIFY Event_Time_Stamps = (the timestamp in step 5, the timestamp
 in step 20, the timestamp in step 12)

25. IF (the object being tested is a multi-state object that supports intrinsic reporting and Protocol_Revision is present and Protocol_Revision \geq 1) THEN
 VERIFY Reliability = MULTI_STATE_FAULT
26. 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)
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-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
29. TRANSMIT BACnet-SimpleACK-PDU
30. IF (the object being tested is NOT an Event Enrollment object) THEN
 VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
31. VERIFY Event_State = NORMAL
32. IF (Protocol_Revision is present and Protocol_Revision \geq 1) THEN
 VERIFY Event_Time_Stamps = (the timestamp in step 5, the timestamp in step 20, the timestamp in step 28)

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 16 can have a value that indicates an unspecified time or a time that precedes the timestamp in step 5.

Another issue is in step 2, that in the case of Event Enrollment, the referenced object is required to have non empty Alarm_Values. This requirement should be on the Event_Parameters of the event_enrollment-Object instead for this case.

wID0241_v6 version of 8.4.2 CHANGE_OF_STATE Test

Purpose: To verify the correct operation of the CHANGE_OF_STATE event algorithm. This test applies to Event Enrollment objects with an Event_Type of CHANGE_OF_STATE and to intrinsic event reporting objects *which generate notifications with an 'Event Type' of CHANGE_OF_STATE*.

Test Concept: The object begins the test in a NORMAL state. The Present Value (referenced property) is changed to a value that is *equal to an alarm value. The definition of an alarm value is specific to the object type. The alarm value shall be not a fault value.* After the time delay *has elapsed*, 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.

Configuration Requirements: *The IUT shall be configured with an object that can detect alarm conditions and send notifications with an Event_Type of CHANGE_OF_STATE. The Event_Detection_Enable property of the object shall be TRUE. The Event_Algorithm_Inhibit property of the object, if present, shall be FALSE.* The Event_Enable property has a value of TRUE for the TO_OFFNORMAL and TO-NORMAL transition. The 'Issue Confirmed Notifications' *parameter in the Recipient List of the configured Notification Class* shall have a value of TRUE. The event-generating object 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 *pMonitoredValue*.

Test Steps:

1. VERIFY *pCurrentState* = NORMAL
2. IF (the object, or *Event_Parameters* if using Event Enrollment, *has* a non-empty *pAlarmValues*) THEN
 3. IF (Present_Value is writable) THEN

WRITE Present_Value = (a value x: x = *pAlarmValues*)

ELSE

MAKE (Present_Value have a value x: x = *one of the pAlarmValues*)
 4. WAIT (*pTimeDelay*)
 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' = (T1, the current local time <i>or</i>
<i>sequence number</i>),	
	'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' =                            (pMonitoredValue, pStatusFlags)

6.      TRANSMIT BACnet-SimpleACK-PDU
7.      IF (the object being tested is NOT an Event Enrollment object) THEN
            VERIFY pStatusFlags = (TRUE, FALSE, ?, ?)
8.      VERIFY pCurrentState = OFFNORMAL
9.      VERIFY Event_Time_Stamps = (T1, *, *)
10.     MAKE (Present_Value have a value x: x corresponds to a NORMAL state)
11.     WAIT (pTimeDelayNormal)
12.     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' =                       (T2, the current local time or sequence
number),
                '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' =                    (pMonitoredValue, pStatusFlags)

13.     TRANSMIT BACnet-SimpleACK-PDU
14.     IF (the object being tested is NOT an Event Enrollment object) THEN
            VERIFY pStatusFlags = (FALSE, FALSE, ?, ?)
15.     VERIFY pCurrentState = NORMAL
16.     IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
            VERIFY Event_Time_Stamps = (T1, *, T2)
17. IF (the object being tested is a multi-state object that supports intrinsic reporting) THEN
18.     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)

```

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' = (T3, 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. IF (the object being tested is NOT an Event Enrollment object) THEN
- VERIFY Status_Flags = (~~TRUE~~FALSE, TRUE, ?, ?)
22. VERIFY Event_State = FAULT
23. IF (Protocol_Revision is present and Protocol_Revision \geq 1) THEN
- VERIFY Event_Time_Stamps = (T1, T3, T2)
24. IF (the object being tested is a multi-state object that supports intrinsic reporting and Protocol_Revision is present and Protocol_Revision \geq 1) THEN
- VERIFY Reliability = MULTI_STATE_FAULT
25. 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)
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' = (T4, 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. IF (the object being tested is NOT an Event Enrollment object) THEN
 VERIFY Status_Flags = (FALSE, FALSE, ?, ?)
 29. VERIFY Event_State = NORMAL
 30. IF (Protocol_Revision is present and Protocol_Revision \geq 1) THEN
 VERIFY Event_Time_Stamps = (T1, T3, T4)

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 T1.

The same problems apply for 8.5.2, though that references 8.4.2 and does not itself represent the steps.

8.5.2 CHANGE_OF_STATE Tests

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

BACnet Reference Clauses: 12.6, 12.8, 12.18, 12.20, 13.2, 13.3.2, and 13.9.

Purpose: To verify the correct operation of the CHANGE_OF_STATE event algorithm. This test 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.

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

Test Steps: The test steps for this test case are identical to the test steps in 8.4.2 except that the ConfirmedEventNotification requests are UnconfirmedEventNotification requests and the TD does not acknowledge receiving the notifications.

Notes to Tester: The passing results for this test case are identical to the ones in 8.4.2 except that the event notifications shall be conveyed using an UnconfirmedEventNotification service request. The MAC address

used for these messages shall be either a broadcast that reaches the local network of the TD or the MAC address of the TD.

Question:

Should the tests be changed to correct these concerns?

Response:

Yes. The indentation will be adjusted as required to match what was in BTL Specified Tests-12.0-final. The manner in which Alarms_Values in step 2 are referred does need to be addressed, and it will be addressed in wID0241.