Clarification Request

References: “e.g” Specified Tests 18.1_final - 7.3.1.11

Date of BTL-WG Response: July 22, 2021

Background: “e.g” Specified Tests 18.1_final - 7.3.1.11, 135.1-2009f-1, 135-2020 - Table 13-3

```
1. VERIFY pCurrentState = NORMAL
2. VERIFY Acked_Transitions = (TRUE, TRUE, TRUE)
3. IF (Protocol_Revision is present AND Protocol_Revision >= 13) THEN
        VERIFY pStatusFlags = (FALSE, FALSE, ?, ?)
4. IF (pMonitoredValue is writable) THEN
        WRITE pMonitoredValue = (a value that is OFFNORMAL)
ELSE
        MAKE (pMonitoredValue have a value that is OFFNORMAL)
5. WAIT (pTimeDelay)
6. BEFORE Notification Fail Time
        RECEIVE ConfirmedEventNotification-Request,
                'Process Identifier' = (PI1: any valid process ID),
                'Initiating Device Identifier' = IUT,
                'Event Object Identifier' = (the event-generating object configured for this test),
                'Time Stamp' = (Toffnormal: any valid time stamp),
                'Notification Class' = (the class corresponding to the object being tested),
                'Priority' = (Poffnormal: the value configured to correspond to a TO-OFFNORMAL
                transition),
                'Event Type' = (any valid event type),
                'Message Text' = (optional, any valid message text),
                'Notify Type' = (the notify type configured for this event),
                'AckRequired' = TRUE,
                'From State' = NORMAL,
                'To State' = OFFNORMAL,
                'Event Values' = (values appropriate to the event type)
7. TRANSMIT BACnet-SimpleACK-PDU
8. VERIFY pCurrentState = OFFNORMAL
9. VERIFY Acked_Transitions = (FALSE, TRUE, TRUE)
10. IF (Protocol_revision is present AND Protocol_Revision >= 13 THEN
        VERIFY pStatusFlags = (TRUE, FALSE, ?, ?)
11. IF (pMonitoredValue is writable) THEN
        WRITE pMonitoredValue = (a value that is NORMAL)
ELSE
        MAKE (pMonitoredValue have a value that is NORMAL)
12. WAIT (pTimeDelayNormal)
13. BEFORE Notification Fail Time
        RECEIVE ConfirmedEventNotification-Request,
                'Process Identifier' = (PI2: any valid process ID),
                'Initiating Device Identifier' = IUT,
                'Event Object Identifier' = (the event-generating object configured for this test),
                'Time Stamp' = (Tnormal: any valid time stamp),
                'Notification Class' = (the class corresponding to the object being tested),
                'Priority' = (Pnormal: the value configured to correspond to a TO-NORMAL transition),
                'Event Type' = (any valid event type),
                'Message Text' = (optional, any valid message text),
                'Notify Type' = (the notify type configured for this event),
        'AckRequired' = TRUE,
        'From State' = OFNORMAL,
        'To State' = NORMAL,
        'Event Values' = (values appropriate to the event type)
14. TRANSMIT BACnet-SimpleACK-PDU
15. VERIFY pCurrentState = NORMAL
16. VERIFY Acked_Transitions = (FALSE, TRUE, FALSE)
17. IF (Protocol_Revision is present AND Protocol_Revision >= 13) THEN
    VERIFY pStatusFlags = (FALSE, FALSE, ?,?)
18. IF (the event-triggering object can be placed into a fault condition) THEN {
    MAKE (a condition exist that will cause the object to generate a fault condition)
    BEFORE **Notification Fail Time**
        RECEIVE ConfirmedEventNotification-Request,
            'Process Identifier' = (PI3: any valid process ID),
            'Initiating Device Identifier' = IUT,
            'Event Object Identifier' = (the event-generating object configured for this test),
            'Time Stamp' = (Tfault: any valid time stamp),
            'Notification Class' = (the class corresponding to the object being tested),
            'Priority' = (Pfault: the value configured to correspond to a TO-FAULT
            transition),
            'Event Type' = IF (Protocol_Revision < 13) THEN
                (any valid event type),
            ELSE
                CHANGE_OF_RELIABILITY,
            'Message Text' = (optional, any valid message text),
            'Notify Type' = (the notify type configured for this event),
            'AckRequired' = TRUE,
            'From State' = NORMAL,
            'To State' = FAULT,
            'Event Values' = (values appropriate to the event type)
    TRANSMIT BACnet-SimpleACK-PDU
    VERIFY pCurrentState = FAULT
    VERIFY Acked_Transitions = (FALSE, FALSE, FALSE)
    TRANSMIT AcknowledgeAlarm-Request,
        'Acknowledging Process Identifier' = (PI3),
        'Event Object Identifier' = (the event-generating object configured for this test),
        'Event State Acknowledged' = FAULT,
        'Acknowledgement Source' = (a character string),
        'Time Stamp' = (Tfault),
        'Time of Acknowledgment' = (the TD’s current time)
    RECEIVE BACnet-SimpleACK-PDU
    IF (Protocol_Revision is present AND Protocol_Revision ≥ 1) THEN
        BEFORE **Notification Fail Time**
            RECEIVE ConfirmedEventNotification-Request,
                'Process Identifier' = (PI3),
                'Initiating Device Identifier' = IUT,
                'Event Object Identifier' = (the event-generating object configured for
                this test),
                'Time Stamp' = (~~Tfault~~ *the IUT’s current time or sequence number*),
                'Notification Class' = (the class corresponding to the object being
                tested),
                'Priority' = (Pfault),
                'Event Type' = IF (Protocol_Revision < 13)
                    (any valid event type),
                ELSE
```

```
                    CHANGE_OF_RELIABILITY,
                'Message Text' = (optional, any valid message text),
                'Notify Type' = ACK_NOTIFICATION,
                'To State' = FAULT
    ELSE
        BEFORE Notification Fail Time
            RECEIVE ConfirmedEventNotification-Request,
                'Process Identifier' = (PI3),
                'Initiating Device Identifier' = IUT,
                'Event Object Identifier' = (the event-generating object configured for this test),
                'Time Stamp' = (~~Tfault~~ the IUT's current time or sequence number),
                'Notification Class' = (the class corresponding to the object being tested),
                'Priority' = (Pfault),
                'Event Type' = (any valid event type),
                'Notify Type' = ACK_NOTIFICATION
    TRANSMIT BACnet-SimpleACK-PDU
    VERIFY Acked_Transitions = (FALSE, TRUE, FALSE)
}
19. TRANSMIT AcknowledgeAlarm-Request,
    'Acknowledging Process Identifier' = (PI2),
    'Event Object Identifier' = (the event-generating object configured for this test),
    'Event State Acknowledged' = NORMAL,
    'Time Stamp' = (Tnormal),
    'Acknowledgement Source' = (a character string),
    'Time of Acknowledgment' = (the TD's current time)
20. RECEIVE BACnet-SimpleACK-PDU
21. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
    BEFORE Notification Fail Time
        RECEIVE ConfirmedEventNotification-Request,
            'Process Identifier' = (PI2),
            'Initiating Device Identifier' = IUT,
            'Event Object Identifier' = (the event-generating object configured for this test),
            'Time Stamp' = (~~Tnormal~~ the IUT's current time or sequence number),
            'Notification Class' = (the class corresponding to the object being tested),
            'Priority' = (Pnormal),
            'Event Type' = (any valid event type),
            'Notify Type' = ACK_NOTIFICATION,
            'To State' = NORMAL
ELSE
    BEFORE Notification Fail Time
    RECEIVE ConfirmedEventNotification-Request,
        'Process Identifier' = (PI2),
        'Initiating Device Identifier' = IUT,
        'Event Object Identifier' = (the event-generating object configured for this test),
        'Time Stamp' = (~~Tnormal~~ the IUT's current time or sequence number),
        'Notification Class' = (the class corresponding to the object bind tested),
        'Priority' = (Pnormal),
        'Event Type' = (any valid event type),
        'Message Text' = (optional, any valid message text),
        'Notify Type' = ACK_NOTIFICATION
22. TRANSMIT BACnet-SimpleACK-PDU
23. VERIFY Acked_Transitions = (FALSE, TRUE, TRUE)
24. TRANSMIT AcknowledgeAlarm-Request,
    'Acknowledging Process Identifier' = (PI1),
    'Event Object Identifier' = (the event-generating object configured for this test),
    'Event State Acknowledged' = OFFNORMAL,
    'Time Stamp' = (Toffnormal),
    'Acknowledgement Source' = (a character string),
    'Time of Acknowledgment' = (the TD's current time)
25. RECEIVE BACnet-SimpleACK-PDU
26. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
    BEFORE **Notification Fail Time**
        RECEIVE ConfirmedEventNotification-Request,
            'Process Identifier' = (PI1),
            'Initiating Device Identifier' = IUT,
            'Event Object Identifier' = (the event-generating object configured for this test),
            'Time Stamp' = (~~Toffnormal~~ *the IUT's current time or sequence number*),
            'Notification Class' = (the class corresponding to the object being tested),
            'Priority' = (Poffnormal),
            'Event Type' = (any valid event type),
            'Message Text' = (optional, any valid message text),
            'Notify Type' = ACK_NOTIFICATION,
            'To State' = OFFNORMAL
ELSE
    BEFORE **Notification Fail Time**
    RECEIVE ConfirmedEventNotification-Request,
            'Process Identifier' = (PI1),
            'Initiating Device Identifier' = IUT,
            'Event Object Identifier' = (the event-generating object configured for this test),
            'Time Stamp' = (~~Toffnormal~~ *the IUT's current time or sequence number*),
            'Notification Class' = (the class corresponding to the object being tested),
            'Priority' = (Poffnormal),
            'Event Type' = (any valid event type),
            'Message Text' = (optional, any valid message text),
            'Notify Type' = ACK_NOTIFICATION
27. TRANSMIT BACnet-SimpleACK-PDU
28. VERIFY Acked_Transitions = (TRUE, TRUE, TRUE)
```

Problem:

These test steps seem to incorrectly/inconsistently indicate that the expected 'Time Stamp' in an ACK_NOTIFICATION is related to the timestamp of the event being acknowledged. This should be updated to indicate that the 'Time Stamp' is the time at which the AcknowledgeAlarm service is executed.

This opinion is based upon a previous clarification in 135.1-2009f-1 and the Table 13-3 of 135-2020.

Question:

Can you review the relevant portions of the standard and update the test steps to expect a proper value for the 'Time Stamp' parameter of the ack notification?

(The highlighted text above are just my suggestions on how to make this more clear. Please change as needed.)

Response:

Yes. The proposed changes are correct.