

Clarification Request

References: 135.1 - 2007 - 9.10.1.2 Unconfirmed COV Notifications, Addendum 135-2008h-5

Background / Proposed Solution:

Addendum 135-2008h-5 Protocol_Revision 10, added an error specific situation table, to Clause

13.14.2 SubscribeCOV service procedure

...

If a new context is created, or a re-subscription is received, then the COV context shall be initialized and given a lifetime as specified by the 'Lifetime' parameter, if present, or zero if the 'Lifetime' parameter is not present. The subscription shall be automatically cancelled after that many seconds have elapsed unless a re-subscription is received. A lifetime of zero shall indicate that the subscription is indefinite and no automatic cancellation shall occur. In either case, a 'Result(+)' shall be returned. A ConfirmedCOVNotification or UnconfirmedCOVNotification shall be issued as soon as possible after the successful completion of a subscription or re-subscription request, as specified by the 'Issue Confirmed Notifications' parameter.

The 'Error Class' and 'Error Code' to be returned for specific situations are as follows:

<i>Situation:</i>	<i>Error Class:</i>	<i>Error Code:</i>
<i>Specified object does not exist</i>	<i>OBJECT</i>	<i>UNKNOWN_OBJECT</i>
<i>Specified object does not support COV notifications</i>	<i>OBJECT</i>	<i>OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED</i>
<i>No context can be created due to resource limitations</i>	<i>RESOURCES</i>	<i>NO_SPACE_TO_ADD_LIST_ELEMENT</i>
<i>The Lifetime parameter is out of the range supported by the device</i>	<i>SERVICES</i>	<i>VALUE_OUT_OF_RANGE</i>

[And adds new Clause 18.2.X, p. 391] defining
OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED

18.2.X OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED – The requested action cannot be executed because the specified object does not support the optional functionality required.

The BTL Specified test 9.10.2.1 was derived from 135.1 - 2003 - 9.10.2.1, for the reason: "This test does not provide for the new error code mandated by the newer revision of the BACnet standard." The BTL Specified test 9.10.2.1 was modified to:

```

2.    IF (Protocol_Revision < 10) THEN
        RECEIVE (BACnet-Error PDU,
            Error Class = SERVICES,
            Error Code = SERVICE_REQUEST_DENIED | OTHER |
                COV_SUBSCRIPTION_FAILED) |
        (BACnet-Error PDU,
            Error Class = PROPERTY,
            Error Code = NOT_COV_PROPERTY) |
        (BACnet-Error PDU,
            Error Class = OBJECT,
```

```

        Error Code =  OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED)
    ELSE
        BACnet-Error PDU,
        Error Class =  OBJECT,
        Error Code =  OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED

```

Tests 135.1 - 2007 - 9.10.1.1 and 135.1 - 2007 - 9.10.1.2 have not been similarly derived

9.10.1.1 Confirmed COV Notifications

Purpose: To verify that the IUT correctly responds to a SubscribeCOV request to establish a subscription for confirmed COV notifications. An implementation that supports COV reporting cannot respond with an error for both this test and the test in 9.10.1.2.

Test Steps:

1. TRANSMIT SubscribeCOV-Request,
 - 'Subscriber Process Identifier' = (any valid process identifier),
 - 'Monitored Object Identifier' = (any object supporting COV notifications),
 - 'Issue Confirmed Notifications' = TRUE,
 - 'Lifetime' = (any value > 0 if automatic cancellation is supported, otherwise 0)
2. RECEIVE BACnet-SimpleACK-PDU
3. WAIT Notification Fail Time
4. IF (the IUT supports confirmed notifications) THEN
 - RECEIVE ConfirmedCOVNotification-Request,
 - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
 - 'Initiating Device Identifier' = IUT,
 - 'Monitored Object Identifier' = (the same object used in the subscription),
 - 'Time Remaining' = (any value > 0 if automatic cancellation is supported, otherwise 0),
 - 'List of Values' = (values appropriate to the object type of the monitored object)
 - ELSE
 - RECEIVE BACnet-Error PDU,
 - Error Class = SERVICES,
 - Error Code = SERVICE_REQUEST_DENIED | OTHER

9.10.1.2 Unconfirmed COV Notifications

Purpose: To verify that the IUT correctly responds to a SubscribeCOV request to establish a subscription for unconfirmed COV notifications. An implementation that supports COV reporting cannot respond with an error for both this test and the test in 9.10.1.1.

Test Steps:

1. TRANSMIT SubscribeCOV-Request,
 - 'Subscriber Process Identifier' = (any valid process identifier),
 - 'Monitored Object Identifier' = (any object supporting COV notifications),
 - 'Issue Confirmed Notifications' = FALSE,
 - 'Lifetime' = (any value > 0 if automatic cancellation is supported, otherwise 0)
2. RECEIVE BACnet-SimpleACK-PDU
3. WAIT Notification Fail Time
4. IF (the IUT supports confirmed notifications) THEN
 - RECEIVE UnconfirmedCOVNotification-Request,

```

        'Subscriber Process Identifier' = (the same identifier used in the
subscription),
        'Initiating Device Identifier' = IUT,
        'Monitored Object Identifier' = (the same object used in the
subscription),
        'Time Remaining' = (any value > 0 if automatic cancellation
is supported, otherwise 0),
        'List of Values' = (values appropriate to the object
type of the monitored object)
    ELSE
        RECEIVE BACnet-Error PDU,
            Error Class = SERVICES,
            Error Code = SERVICE_REQUEST_DENIED | OTHER

```

Question:

Does test 9.10.1.1 want to be similar adapted, to adopt the more permissive success criteria like that of BTL Specified test 9.10.2.1 shown above, when the Monitored Object of Device Does Not Support ConfirmedCOVNotification? Similarly does test 9.10.1.2 want to be similar adapted, to adopt the more permissive success criteria of BTL Specified test 9.10.2.1, when the Monitored Object of Device Does Not Support UnconfirmedCOVNotification?

Response:

No, tests 9.10.1.1 and 9.10.1.2 are positive tests. The Test Plan should be filtering on this. The tests would not be executed against objects which will not accept the subscription. The BTL-WG notes that the current form of the test in step 4 is incorrect, and should not allow an error response from the IUT.