

## Clarification Request

**References:** BTL Test Plan 9.0.final, 135.1-2009 test 9.7.2.3

**Date of BTL-WG Response:** June 5, 2012

### Background:

#### 135.1-2009 test 9.7.2.3

##### 9.7.2.3 Event Type Filter

Purpose: To verify that the IUT can execute the GetEnrollmentSummary request when the 'Event Type Filter' is used.

Configuration Requirements: If possible, the IUT shall be configured so that it has one or more event-generating objects for each of the event types CHANGE\_OF\_BITSTRING, CHANGE\_OF\_STATE, CHANGE\_OF\_VALUE, COMMAND\_FAILURE, FLOATING\_LIMIT, and OUT\_OF\_RANGE. If only a subset of these event types are supported as many of them as possible shall be configured.

Test Steps:

1. TRANSMIT GetEnrollmentSummary-Request,  
'Acknowledgment Filter' = ALL,  
'Event Type Filter' = CHANGE\_OF\_BITSTRING
2. RECEIVE GetEnrollmentSummary-ACK,  
'List of Enrollment Summaries' = (all configured event-generating objects with Event\_Type = CHANGE\_OF\_BITSTRING)
3. TRANSMIT GetEnrollmentSummary-Request,  
'Acknowledgment Filter' = ALL,  
'Event Type Filter' = CHANGE\_OF\_STATE
4. RECEIVE GetEnrollmentSummary-ACK,  
'List of Enrollment Summaries' = (all configured event-generating objects with Event\_Type = CHANGE\_OF\_STATE)
5. TRANSMIT GetEnrollmentSummary-Request,  
'Acknowledgment Filter' = ALL,  
'Event Type Filter' = CHANGE\_OF\_VALUE
6. RECEIVE GetEnrollmentSummary-ACK,  
'List of Enrollment Summaries' = (all configured event-generating objects with Event\_Type = CHANGE\_OF\_VALUE)
7. TRANSMIT GetEnrollmentSummary-Request,  
'Acknowledgment Filter' = ALL,  
'Event Type Filter' = FLOATING\_LIMIT
8. RECEIVE GetEnrollmentSummary-ACK,  
'List of Enrollment Summaries' = (all configured event-generating objects with Event\_Type = FLOATING\_LIMIT)

Suppose the IUT contains event generating objects where the notification-class property does currently not point to an existing notification class inside the IUT. This situation might have a number of possible reasons:

- a vendor has designated the “wildcard” object instance as indicating that no notification class has currently been assigned (e.G. Immediately after creation of the object).
- the notification class object that had properly been assigned has since been deleted in the IUT.

Now the GetEnrollmentSummary-ACK needs to provide a “priority” which is supposed to be taken from the assigned notification-class object. How should the IUT obtain that priority value?

I found two other documents that might provide helpful information:

#### **BTL-CRR-0033**

There is a statement, that the only acceptable reason an object should not show up in the GetEnrollmentSummary response should be False event-enable flags for all three transitions.

In the response there is a sentence, stating that “misconfigured” products should not be tested.

The question is whether above situation should be considered “misconfigured”. As this situation can be created with normal BACnet means and can be expected to occur in normal operation I think there should be rules how the IUT is supposed to behave in the test.

#### **IC 135-2004-26**

there is a mention of “not fully configured” objects which are then supposed not to be able to generate notifications.

#### **Possible solutions:**

- do not include objects in the answer, that do not point to a valid nc object.
- Use some arbitrary priority value (like 0 or ???) in the answer.

#### **Additional Background:**

9.7.2.3 has already been discussed in CR-0197 and is part of wID0078.

#### **Question:**

How should the above described condition be handled in the test?

#### **Response:**

The BTL is not going to rule on how devices should respond when there are objects within them that are not fully configured. There are known issues with this, that have been dealt with in Protocol\_Revision 13 and our efforts will be concentrated on that Protocol\_Revision.