

## Clarification Request

**References:** “e.g.” BTL Specified Tests-15.2.final.2

**Date of BTL-WG Response:** July 11, 2019

☒ All Actions Necessitated have been Completed

**Background:** BTL - 7.3.1.X8.1 - Reliability\_Evaluation\_Inhibit Test

### 7.3.1.X8.1 Reliability\_Evaluation\_Inhibit Test

Reason for Change: New functionality added with Addendum 135-2010af. This test does not exist in 135.1-2013.

Purpose: To verify that Reliability\_Evaluation\_Inhibit controls whether or not fault conditions are detected.

Test Concept: Select an event generating object, O1, which supports the Reliability\_Evaluation\_Inhibit property. With Reliability\_Evaluation\_Inhibit FALSE, make a fault condition exist. Verify that Reliability changes and that a notification is generated. Set Reliability\_Evaluation\_Inhibit to TRUE. Verify that the Reliability changes to NO\_FAULT\_DETECTED and that a TO\_NORMAL notification is generated. Remove the fault condition and ensure that no notification is generated. Make a fault condition exist and verify that Reliability remains NO\_FAULT\_DETECTED, and that no notification is generated.

Test Configuration: O1 is configured to detect and report unconfirmed events, is in the NORMAL state, and Reliability\_Evaluation\_Inhibit equals FALSE, so that reliability evaluation for that object is configured to detect fault conditions. **If no object exists in the IUT for which fault conditions can be generated, then this test shall be skipped.**

1. VERIFY Event\_State = NORMAL
2. VERIFY Reliability = NO\_FAULT\_DETECTED
3. MAKE (a fault condition exist for O1)
4. BEFORE **Notification Fail Time**
  - RECEIVE UnconfirmedEventNotification-Request
    - 'Process Identifier' = (the value configured for the transition),
    - 'Initiating Device Identifier' = IUT,
    - 'Event Object Identifier' = O1,
    - 'Time Stamp' = (any valid timestamp),
    - 'Priority' = (any valid priority),
    - 'Event Type' = CHANGE\_OF\_RELIABILITY,
    - 'Notify Type' = ALARM | EVENT,
    - 'Message Text' = (any valid message text),
    - 'AckRequired' = TRUE | FALSE,
    - 'From State' = NORMAL,
    - 'To State' = FAULT,
    - 'Event Values' = (any values appropriate to CHANGE\_OF\_RELIABILITY)
5. WRITE Reliability\_Evaluation\_Inhibit = TRUE
6. BEFORE **Internal Processing Fail Time + Notification Fail Time**
  - RECEIVE UnconfirmedEventNotification-Request
    - 'Process Identifier' = (the value configured for the transition),
    - 'Initiating Device Identifier' = IUT,
    - 'Event Object Identifier' = O1,

- 'Time Stamp' = (any valid timestamp),  
 'Priority' = (any valid priority),  
 'Event Type' = CHANGE\_OF\_RELIABILITY,  
 'Notify Type' = ALARM | EVENT,  
 'Message Text' = (any valid message text),  
 'AckRequired' = TRUE | FALSE,  
 'From State' = FAULT,  
 'To State' = NORMAL,  
 'Event Values' = (any values appropriate to CHANGE\_OF\_RELIABILITY)
7. VERIFY Reliability = NO\_FAULT\_DETECTED
  8. VERIFY Event\_State = NORMAL
  9. MAKE (remove the fault condition)
  10. WAIT **Notification Fail Time**
  11. CHECK (that the IUT did not send any event notifications for O1)
  12. MAKE (a fault condition exist for O1)
  13. WAIT **Notification Fail Time**
  14. VERIFY Reliability = NO\_FAULT\_DETECTED
  15. VERIFY Event\_State = NORMAL
  16. CHECK (that the IUT did not send any event notifications for O1)

Notes to Tester: This behavior can alternately be tested using the ConfirmedEventNotification service, but it is not necessary to test both.

#### Problem:

We have one claim “**Contains an object with Reliability\_Evaluation\_Inhibit Property**” under each object, which gets added if the Reliability\_Evaluation\_Inhibit Property is present.

Above mentioned testcase is applicable under that claim.

We get devices in lab, which has writable Reliability property writable, and after writing its value other than No\_Fault\_Detected, the IUT initiates fault notification with Event\_Type as Change\_Of\_Reliability.

Now below the different questions around this,

#### Question:

1. If the mechanism of making an object enter in fault condition is by writing the Reliability property will BTL - 7.3.1.X8.1 testcase be applicable?
2. Or based on the statement mentioned in the Test Configuration, ***‘If no object exists in the IUT for which fault conditions can be generated then this test shall be skipped’***, will this testcase be skipped?
3. If testcase is applicable step number 14 needs to be modified as Reliability will not have value NO\_FAULT\_DETECTED, since writing Reliability is the only way to make the device enter in Fault condition.

#### Response:

- 1) NO
- 2) n/a See question 1.
- 3) n/a See question 1.