

Clarification Request

References: Specified Tests 15.2.final.2 - 7.3.1.X8.1 and 7.3.1.X8.2

Date of BTL-WG Response: 19-Sept-2019

☒ All Actions Necessitated have been Completed

Background: TP 15.2 Specified Tests 7.3.1.X8.1 and 7.3.1.X8.2 require optional functionality

BTL - 7.3.1.X8.1 - Reliability_Evaluation_Inhibit Test

Test Method Manual

Configuration As per ***BTL Specified Tests***.

Test Conditionality If no object exists in the IUT for which fault conditions can be generated then this test shall be skipped.

Test Directives

Testing Hints

Notes & Results

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.

Test Steps:

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.

BTL - 7.3.1.X8.2 - Reliability_Evaluation_Inhibit Summarization Test

Test Method

Manual

Configuration

As per *BTL Specified Tests*.

Test Conditionality

If no object exists in the IUT for which fault conditions can be generated then this test shall be skipped.

Test Directives

Testing Hints

Notes & Results

7.3.1.X8.2 Reliability_Evaluation_Inhibit Summarization Test

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

Purpose: To verify that event generating objects are reported by summarization routines as needed even when the reliability evaluation has been inhibited.

Test_Concept: Select an event generating object O1 which is configured for event reporting, which is configured to require acknowledgement for TO_FAULT transition, and its Acked_Transitions property is (T, F, T). Verify that the event is reported when the device responds to a GetEventInformation request.

Configuration Requirements: O1 is configured such that it requires acknowledgement of the TO_FAULT transition, and the Acked_Transitions is (T, F, T). O1's Reliability_Evaluation_Inhibit equals TRUE, so that reliability evaluation for that object is inhibited. The number of event generating objects in the IUT that have an Event_State other than NORMAL, or which have an Acked_Transitions other than (T, T, T) is such that they can all be reported in a single GetEventInformation-ACK response.

Test Steps:

1. VERIFY Acked_Transitions = (T, F, T)

2. VERIFY Reliability_Evaluation_Inhibit = TRUE
3. TRANSMIT GetEventInformation
4. RECEIVE GetEventInformation-Ack,
'List of Event Summaries' = (list of object identifiers which includes O1)
'More Events' = FALSE

Problem:

Test 7.3.1.X8.1 requires the optional functionality to send unconfirmed or confirmed EventNotifications.
Test 7.3.1.X8.2 requires the optional functionality to support GetEventInformation

We have an IUT with objects that have Reliability and Reliability_Evaluation_Inhibit Properties.
The IUT does not support EventNotification and GetEventInformation. According to the standard this is not required for Reliability_Evaluation_Inhibit.

The Test Conditionality only allows to skip the test if no object exists in the IUT for which fault conditions can be generated.

Question:

1. Does test 7.3.1.X8.1 have to be changed so it is able to verify that Reliability_Evaluation_Inhibit controls whether or not fault conditions are detected, without requiring that the IUT send EventNotifications?
2. Is it allowed to skip test 7.3.1.X8.2 if the IUT does not support GetEventInformation?

Response:

1. Yes.
2. Yes.