

Clarification Request

Request From: "Christian Enklaar" <cen@mbs-solutions.de>

References: Interim Test Specification Version 10 for TP 16.1; BACnet Standard 135-2016

Stage: Request, Listed, Analysis, Resolved

Actions Necessitated: Checklist/Test Plan Change, BTL Specified Tests Change,
7.3.2.X43.3
SSPC Interpretation Required, Implementation Guidelines Change,

Date of BTL-WG Response: July 9, 2020

All Actions Necessitated have been Completed

Background:

7.3.2.X43.3Out_Of_Service, Status_Flags, and Reliability test for an Object that does not contain Present_Value

Purpose: This test verifies the interrelationship between the Out_Of_Service, Status_Flags, and Reliability properties. If the PICS indicates that the Out_Of_Service property of the object under test is not writable, and if the value of the property cannot be changed by other means, then this test shall be omitted. This test applies to objects that do not contain Present_Value.

Test Concept: Write to and verify the interrelationship between the Out_Of_Service, Status_Flags, and Reliability properties of an object which does not contain Present_Value.

Configuration Requirements: The selected object is configured such that its Reliability is NO_FAULT_DETECTED before execution of this test.

Test Steps:

1. IF (Out_Of_Service is writable) THEN
 WRITE Out_Of_Service = TRUE
 ELSE
 MAKE (Out_Of_Service = TRUE)
2. VERIFY Out_Of_Service = TRUE
3. VERIFY Status_Flags = (?, FALSE, ?, TRUE)
4. IF (Reliability is present and writable) THEN
 REPEAT X = (all values of the Reliability enumeration appropriate to the object type
 except NO_FAULT_DETECTED) DO {
 WRITE Reliability = X
 VERIFY Reliability = X
 VERIFY Status_Flags = (TRUE, TRUE, ?, TRUE) (? , TRUE, ?, TRUE)
 WRITE Reliability = NO_FAULT_DETECTED
 VERIFY Reliability = NO_FAULT_DETECTED
 VERIFY Status_Flags = (? FALSE, ?, TRUE)
 }
5. CHECK (all communication of the protocol modeled by the object, through that port is disabled)
6. IF (Out_Of_Service is writable) THEN

```
        WRITE Out_Of_Service = FALSE
    ELSE
        MAKE (Out_Of_Service = FALSE)
7.  VERIFY Out_Of_Service = FALSE
8.  VERIFY Status_Flags = ( ?, ?, ?, FALSE)
```

Problem:

If an Object (like Network Port) does not support intrinsic reporting, the standard says:

12.56.5 Status_Flags

...
IN_ALARM Logical TRUE (1) if the Event_State property is present and does not have a value of NORMAL,
otherwise logical FALSE (0).
...

12.56.67 Event_State

...
If the object does not support event reporting, then the value of this property shall be NORMAL.

But if the Event_State remains NORMAL (or the optional property does not exist), the alarm flag will not be set in the Status_Flags and the test would fail.

There already has been an equal CR for the Accumulator Object (BTL-CR-0449_7.3.2.X37.1.6_OutOfService_AccumulatorTest).

Question:

Should the test be changed as outlined above?

Response:

YES.