

Clarification Request

References: 135.1-2007

Background / Proposed Solution:

The test 135.1-2007-7.3.1.1 Out_Of_Service, Status_Flags and Reliability Tests, requires that the object we are testing not be in FAULT in order to successfully match the Status_Flags property.

I believe the object will behave in the same way, since the Out_Of_Service property is set to TRUE during the test and therefore this test could be modified.

This change would allow testing of objects that are in FAULT and do not have a writable Reliability property.

I am also recommending the Test Concept be removed. The Test Plan is used to direct the tester to which object type to use. I also believe the test steps provide sufficient description that if the Reliability is not writable that section of the test shall be skipped.

7.3.1.1 Out_Of_Service, Status_Flags, and Reliability Tests

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: 12.1.7, 12.1.9, 12.1.10, 12.2.7, 12.2.9, 12.2.10, 12.3.7, 12.3.9, 12.3.10, 12.4.6, 12.4.8, 12.4.9, 12.6.7, 12.6.9, 12.6.10, 12.7.7, 12.7.9, 12.7.10, 12.8.6, 12.8.8, 12.8.9, 12.15.8, 12.15.10, 12.15.11, 12.16.8, 12.16.10, 12.16.11, 12.17.6, 12.17.8, 12.17.9, 12.18.7, 12.18.9, 12.18.10, 12.19.7, 12.19.9, 12.19.10, 12.20.6, 12.20.8, 12.20.9, 12.23.7, 12.23.9, and 12.23.10.

Purpose: This test case verifies that Present_Value is writable when Out_Of_Service is TRUE. It also 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 Accumulator, Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value, Life Safety Point, Life Safety Zone, Multi-state Input, Multi-state Output, Multi-state Value, Loop and Pulse Converter objects.

Test Concept: The IUT will select one instance of each appropriate object type and test it as described. If the Reliability property is not supported then step 4 *step 6* shall be omitted.

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 = (?,?,?,TRUE)(?,?,?,TRUE)
4. REPEAT X = (all values meeting the functional range requirements of 7.2.1) DO {
 WRITE Present_Value = X
 VERIFY Present_Value = X
}
5. WRITE Present_Value = (any value that corresponds to an Event_State of NORMAL)
6. IF (Reliability is 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)
 WRITE Reliability = NO_FAULT_DETECTED
 VERIFY Reliability = NO_FAULT_DETECTED
 VERIFY Status_Flags = (?,?,FALSE,TRUE)
 }
7. IF (Out_Of_Service is writable) THEN
 WRITE Out_Of_Service = FALSE
ELSE
 MAKE (Out_Of_Service FALSE)

Notes to Tester: If the object being tested is commandable and there is an internal process writing to the Present_Value property each WriteProperty request shall contain a priority sufficient to override the internal process. After step 4 step 6 the priority array slot shall be relinquished.

Question:

Is this interpretation Correct?

Response:

Yes. The following test changes also include recent proposed changes from DJH-001-3.

7.3.1.1 Out_Of_Service, Status_Flags, and Reliability Tests

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: 12.1.7, 12.1.9, 12.1.10, 12.2.7, 12.2.9, 12.2.10, 12.3.6, 12.3.8, 12.3.9, 12.5.7, 12.5.9, 12.5.10, 12.6.7, 12.6.9, 12.6.10, 12.7.6, 12.7.8, 12.7.9, 12.14.8, 12.14.10, 12.14.11, 12.15.8, 12.15.10, 12.15.11, 12.16.6, 12.16.8, 12.16.9, 12.17.7, 12.17.9, 12.17.10, 12.18.7, 12.18.9, 12.18.10, 12.19.6, 12.19.8, and 12.19.9.

Purpose: This test case verifies that Present_Value is writable when Out_Of_Service is TRUE. It also 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 Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value, Life Safety Point, Life

Safety_Zone, Multi-state Input, Multi-state Output, Multi-state Value, and Loop objects all object types that have an Out_Of_Service property, that when TRUE, results in the Present_Value property being writable.

Test Concept: The IUT will select one instance of each appropriate object type and test it as described. If the Reliability property is not supported then step 4 step 5 shall be omitted.

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. REPEAT X = (all values meeting the functional range requirements of 7.2.1) DO {
 WRITE Present_Value = X
 VERIFY Present_Value = X
}
5. ~~WRITE Present_Value = (any value that corresponds to an Event_State of NORMAL)~~
- 6.5. 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)
 WRITE Reliability = NO_FAULT_DETECTED
 VERIFY Reliability = NO_FAULT_DETECTED
 VERIFY Status_Flags = (?, FALSE, ?, TRUE)
 }
- 7.6. 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)*

Notes to Tester: If the object being tested is commandable and there is an internal process writing to the Present_Value property each WriteProperty request shall contain a priority sufficient to override the internal process. After step 4 the priority array slot shall be relinquished.