

Clarification Request

References: BTL Specified Tests test 8.18.3

Date of BTL-WG Response: April 18, 2017

Background:

In Scheduling - Advanced View Modify - A, there is use of test 135.1-2009g-14 - 8.18.3 with this Testing Hint:

Verify for Weekly_Schedule, Exception_Schedule, Effective_Period , Date_List, Schedule_Default, List_Of_Object_Property_References, Priority_For_Writing, Out_Of_Service properties.

See Addendum L for requirements details.

The reference schedule used during this test should include an Exception_Schedule that contains 255 entries and contain 12 BACnetTimeValue tuples per entry. The reference schedule should also contain a Weekly_Schedule which contains 6 BACnetTimeValue tuples per day. The Calendar Date_List used in this test should contain 32 calendar entries.

8.18.3 Reading and Presenting Properties

Purpose: This test case verifies that the IUT is capable of reading and presenting properties. It is a generic test used to test data presentation requirements.

Configuration: For this test, the tester shall choose a property, P_1 , from an object, O_1 . The TD shall be configured to not support the execution of ReadPropertyMultiple or the initiation of COV notifications.

Test Steps:

1. MAKE (the IUT read P_1)
2. RECEIVE ReadProperty-Request,
 'Object Identifier' = O_1 ,
 'Property Identifier' = P_1
3. TRANSMIT BACnet-ComplexACK-PDU,
 'Object Identifier' = O_1 ,
 'Property Identifier' = P_1 ,
 'Property Value' = (any valid value for P_1)
4. CHECK (that the IUT presents a value that is consistent with the value received in step 3)

Notes to Tester: The value presented by the IUT may differ from the value transmitted on the wire due to rounding, truncation, formatting, language conversion etc.

Notes to Tester: If the property being read and displayed is an array, the IUT may include an Array Index parameter in the ReadProperty-Request in step 2 and the TD will include it in the response in step 4.

Notes to Tester: If the IUT has not already determined that the TD does not support execution of ReadPropertyMultiple, SubscribeCOV, and SubscribeCOVProperty, the IUT may initiate any of these services. If this occurs, the IUT shall pass the test only if it automatically falls back to using ReadProperty upon receipt of the correct BACnetReject-PDU(s) from the TD, indicating that other service(s) is not supported.

It has always been interpreted that that huge Exception_Schedule needs to be read and updated in pieces by the client using access by index, after TD responds with "buffer overflow" when a client is observed reading that Exception_Schedule in its entirety.

A workstation vendor who aborts the operation upon receiving "buffer overflow", claims correct implementation in the TD should respond with "segmentation not supported". This idea originates from test 13.1.12.1, where passing result is: 'SEGMENTATION_NOT_SUPPORTED' as the abort code when IUT returns an abort message that it does not support segmented responses and a response would be larger than 1 segment.

Note though in test 135.1 - 9.18.1.4 (unused in BTL Test Plan) in its Notes to Tester: If the object list is too long to return given the APDU and segmentation limitations of the IUT and TD, an Abort message indicating "segmentation not supported" or "buffer overflow" is a passing result.

Question:

Is it required that a SCHED-AVM-A implementation upon receiving "buffer overflow" as the abort code when attempting to read any huge property value from Schedule in its entirety, needs to fallback and read and update in pieces using access by index?

Response: Neither doing it automatically nor using a fallback mechanism is required. But there must be a way to MAKE the workstation present every value in the mandated example schedule."