Clarification Request

References: BTL Specified Tests 5.0

Background / Proposed Solution:

The Read-only Property test is designed to test that the IUT 1) responds with the correct error message, 2) verifies the EPICS correctly advertises the writable properties in the IUT.

The problem with this test takes a long time to run especially on devices whose database has hundreds if not thousands of objects.

Assume that this test takes 5 seconds to run on 1 property in 1 object if it succeeds. It will take longer to wait on the failure timeout if the IUT does not respond correctly. Also assume each object has about 12 read-only properties. That results in 1 minute per object. With 400 objects, we have 400 minutes or 6 hours to run. Remember if there are incorrect responses this will take longer to run.

7.2.2.1 Read-only Property Test

Reason For Change: There is no test that verifies that properties listed as read-only in the EPICS are in fact read-only. This test is included in the SSPC Proposal CN-112.

Purpose: To verify that properties marked as read-only in the EPICS are in fact read-only.

Test Concept: Write to each read-only (not writable and not conditionally writable) property in the EPICS, the value of the property form the EPICS and verify that an error is returned. Write another value that is within the acceptable range for the datatype and verify that an error is returned. If the property is a list and the IUT supports AddListElement, attempt to modify the property with AddListElement and verify that an error is returned. If the IUT does not support the WriteProperty service, this test shall be skipped.

Test Steps:

1. REPEAT X = (all objects in the IUT's database) DO {
   REPEAT Y = (all read-only properties in object X) DO {
   IF (the property is not an array) THEN
   IF (the property value is provided in the EPICS) THEN
   TRANSMIT WriteProperty-Request,
   'Object Identifier' = X,
   'Property Identifier' = Y,
   'Property Value' = (the value from the EPICS)
   RECEIVE BACnet-Error-PDU,
   Error Class = PROPERTY,
   Error Code = WRITE_ACCESS_DENIED

   TRANSMIT WriteProperty-Request,
   'Object Identifier' = X,
   'Property Identifier' = Y,

```
                'Property Value' =          (any value meeting the range requirements
                                            of 7.2.1 except that value from the EPICS)
        RECEIVE BACnet-Error-PDU,
                Error Class =               PROPERTY,
                Error Code =                WRITE_ACCESS_DENIED

        IF (the IUT supports AddListElement and the property is a list) THEN
                TRANSMIT AddListElement-Request,
                        'Object Identifier' =       X,
                        'Property Identifier' =     Y,
                        List of Elements' =         (any elements value meeting the range
                                                    requirements of 7.2.1 except any in the
                                                    value from the EPICS)
                RECEIVE BACnet-Error-PDU,
                        Error Class =               PROPERTY,
                        Error Code =                WRITE_ACCESS_DENIED
ELSE
        IF (the property value is provided in the EPICS and the array at least 1 element) THEN
                TRANSMIT WriteProperty-Request,
                        'Object Identifier' =       X,
                        'Property Identifier' =     Y,
                        'Property Value' =          (the value from the EPICS)
                        'Array Index' =             1
                RECEIVE BACnet-Error-PDU,
                        Error Class =       PROPERTY,
                        Error Code =        WRITE_ACCESS_DENIED

        TRANSMIT WriteProperty-Request,
                'Object Identifier' =       X,
                'Property Identifier' =     Y,
                'Property Value' =          (any value meeting the range requirements
                                            of 7.2.1 except that value from the EPICS)
                'Array Index' =             1
        RECEIVE BACnet-Error-PDU,
                Error Class =               PROPERTY,
                Error Code =                WRITE_ACCESS_DENIED

        IF (the IUT supports AddListElement and the property is a list) THEN
                TRANSMIT AddListElement-Request,
                        'Object Identifier' =       X,
                        'Property Identifier' =     Y,
                        'Array Index' =             1
                        List of Elements' =         (any elements value meeting the range
                                                    requirements of 7.2.1 except any in the
                                                    value from the EPICS)
                RECEIVE BACnet-Error-PDU,
                        Error Class =               PROPERTY,
                        Error Code =                WRITE_ACCESS_DENIED
    }
}
```

Notes to tester: When modifying a property, it is expected that an error class of PROPERTY with an error code of WRITE_ACCESS_DENIED is returned but the IUT may instead return an error_class of PROPERTY with an error_code of VALUE_OUT_OF_RANGE, or an error_class of RESOURCES with an error_code of NO_SPACE_TO_WRITE_PROPERTY. In the case that the property is an array, and it

has no elements, then the IUT may return and error class of PROPERTY and an error code of INVALID_ARRAY_INDEX.

Question:

I propose we re-write the test to allow testing of one or two unique objects of each object type in the IUT database to be selected by the tester. The test plan should be changed to provide instruction to select one or two instances of each object type. The test should then be modified to remove the 'repeat all objects in the IUT database'.

Response:

Add this section:

Notes to Tester: The objects selected by the tester should include one instance of each supported object type. Where some instances of an object type differ in the set of supported properties, the allowable value ranges for a property, or the writability of a property, then one instance of each “flavor” of that object type should be selected.

and change the first part of Step 1 as here indicated:

1. REPEAT X = (~~all objects in the IUT's database~~ *a tester selected set of objects*) DO {
   REPEAT Y = (all read-only properties in object X) DO {
   IF (the property is not an array) THEN