

## Clarification Request

**References:** ASHRAE 135.1-2023, ASHRAE 135-2024, Test Package 26.0.

**Date of BTL-WG Response:** May 29, 2025

**Background:**

---

## 4.6 Data Sharing - WriteProperty - B

---

### 4.6.1 Base Requirements

Base requirements must be met by any IUT claiming conformance to this BIBB.

...		
<b>BTL - 9.22.1.X3 - Writing NULL to Non-commandable Properties</b>		
	<b>Test Conditionality</b>	If the IUT claims Protocol_Revision 20, or prior, this test shall be skipped. If the IUT does not contain any writable non-commandable properties, this test shall be skipped.
	<b>Test Directives</b>	Repeat the test for a selection of <b>writable non-commandable, non-Present Value properties</b> which do not support the value NULL. Repeat the test for each object type with a writable non-commandable Present_Value supported by the IUT which do not support the value NULL.
	<b>Testing Hints</b>	

#### 9.22.1.X3 Writing NULL to Non-commandable Properties

Reason for Change: The standard was changed in PR21 to require that devices not return errors when a NULL is written to non-commandable properties and no test exists for this functionality.

Purpose: This test case verifies that the IUT returns a Result(+) when an attempt is made to relinquish a non-commandable property.

Test Concept: Write NULL to a writable non-commandable property, P1 in object O1, and verify the IUT returns a Result(+) and does not modify the property.

Test Configuration: P1 shall be a property for which NULL is not an accepted value.

Test Steps:

1. READ X = (O1), P1
2. TRANSMIT WriteProperty-Request,  
    'Object Identifier' = O1,  
    'Property Identifier' = P1,  
    'Property Value' = NULL
3. RECEIVE BACnet-SimpleACK-PDU
4. VERIFY (O1), P1 = X

#### 15.9.2 Service Procedure

...

If an attempt is made to relinquish a property that is not commandable and for which Null is not a supported datatype, if no other error conditions exist, the property shall not be changed, and the write shall be considered successful. See Clause 19.

The standard uses the term 'relinquish' to mean to write a Null at a priority to a commandable property and only the Present\_Value property can be commandable.

From an interoperability point of view, the main benefit of this capability is to allow a client to not have to determine if a value object is commandable before relinquishing or not having to deal with error codes if the object is not commandable. There is also no interoperable benefit in requiring a device to accept a relinquish to the High\_Limit property.

**Problem:**

Test 9.22.1.X3 requires a device that receives a write request, that contains a NULL, destined for any writable non-commandable property to accept the request but not process the write.

Based on Clause 15.9.2, does the standard require writing with a Null to all non-commandable properties? Plus, what does "...if no other error conditions exist..." actually mean?

**Question:**

Can test 9.22.1.X3 be limited to relinquishing the Present\_Value property in non-commandable value objects otherwise this test can be skipped?

**Response:**

**Yes**