Clarification Request

Request from: BTL Manager <btl-manager@bacnetinternational.org>

References: 135.1-2009 and BTL Specified Tests-9.0.final

Stage: ☒Request, ☐Listed, ☐Analysis, ☐Resolved

Actions necessitated: ☐Checklist/Test Plan change, ☒BTL Specified Tests change,
Section _____, Test number: _9.18.2.3. and 9.20.1.6_
☐SSPC Interpretation required, ☐Implementation Guidelines change,

Date of BTL-WG Response: _____
☐All actions necessitated have been completed

Background / Proposed Solution:

135.1-2009 is the source for test 9.18.2.3

BTL Specified Tests-9.0.final contains test 9.20.1.6 which was made more lenient regarding UNSUPPORTED_OBJECT_TYPE.

135.1-2009l-11, recently approved for publication and included in the republication of ASHRAE 135.1-2011, revises test 9.20.1.6, without clarifying the asymmetry regarding UNSUPPORTED_OBJECT_TYPE.

135.1.-2009 - 9.18.2.3 Reading an Unknown Object
Purpose: To verify that the IUT can execute ReadProperty service requests under circumstances where the requested object does not exist.

Test Concept: The TD attempts to read a property that is not defined for the specified object.

Test Steps:

1. TRANSMIT ReadProperty-Request,
'Object Identifier' = (any standard object not contained in the IUT's database),
'Property Identifier' = (any property defined for the specified object)
2. RECEIVE BACnet-Error-PDU,
Error Class = OBJECT,
Error Code = UNKNOWN_OBJECT

BTL Specified Tests 9.20.1.6 Reading Multiple Properties with Multiple Embedded Access Errors

Reason For Change: The BTL decided that, depending on the parameters of the test, the IUT could also return UNSUPPORTED_OBJECT_TYPE for properties in Object 2. This test is not in any SSPC proposal.

Purpose: To verify the ability to correctly execute a ReadPropertyMultiple service request for which the 'List of Read Access Specifications' contains specifications for multiple unsupported properties.

Test Steps:

1. TRANSMIT ReadPropertyMultiple-Request,
   'Object Identifier' = Object1,
   'Property Identifier' = P1,
   'Property Identifier' = P2,
   'Property Identifier' = (any property, P3, not supported in this object),
   'Property Identifier' = (any property, P4, not supported in this object),
   'Object Identifier' = (any object, Object2, not supported in the IUT)
   'Property Identifier' = P5,
   'Property Identifier' = P6
2. RECEIVE ReadPropertyMultiple-ACK,
   'Object Identifier' = Object1,
   'Property Identifier' = P1,
   'Property Value' = (the value of P1 specified in the EPICS),
   'Property Identifier' = P2,
   'Property Value' = (the value of P2 specified in the EPICS),
   'Property Identifier' = P3,
   'Error Class' = PROPERTY,
   'Error Code' = UNKNOWN_PROPERTY,
   'Property Identifier' = P4,
   'Error Class' = PROPERTY,
   'Error Code' = UNKNOWN_PROPERTY,
   'Object Identifier' = Object2,
   'Property Identifier' = P5,
   'Error Class' = OBJECT,
   'Error Code' = (UNKNOWN_OBJECT|UNSUPPORTED_OBJECT_TYPE),
   'Property Identifier' = P6,
   'Error Class' = OBJECT,
   'Error Code' = (UNKNOWN_OBJECT|UNSUPPORTED_OBJECT_TYPE)

135.1.-2009l-11 9.20.1.6 Reading Multiple Properties with Multiple Embedded Access Errors

Purpose: To verify the ability to correctly execute a ReadPropertyMultiple service request for which the 'List of Read Access Specifications' contains specifications for multiple unsupported properties.

Test Steps:

1. TRANSMIT ReadPropertyMultiple-Request,
   'Object Identifier' = O1,
   'Property Identifier' = P1,
   'Property Identifier' = P2,
   'Property Identifier' = (any property, P3, not supported in this object),
   'Property Identifier' = (any property, P4, not supported in this object),
   'Object Identifier' = (any *non-existent* object, O2, *which is of a type supported by the IUT),* ~~not supported in the IUT)~~
   'Property Identifier' = P5,
   'Property Identifier' = P6
2. RECEIVE ReadPropertyMultiple-ACK,
   'Object Identifier' = O1,
   'Property Identifier' = P1,
   'Property Value' = (the value of P1 specified in the EPICS),

'Property Identifier' =	P2,
'Property Value' =	(the value of P2 specified in the EPICS),
'Property Identifier' =	P3,
'Error Class' =	PROPERTY,
'Error Code' =	UNKNOWN_PROPERTY,
'Property Identifier' =	P4,
'Error Class' =	PROPERTY,
'Error Code' =	UNKNOWN_PROPERTY,
'Object Identifier' =	O2,
'Property Identifier' =	P5,
'Error Class' =	OBJECT,
'Error Code' =	(UNKNOWN_OBJECT),
'Property Identifier' =	P6,
'Error Class' =	OBJECT,
'Error Code' =	(UNKNOWN_OBJECT)

Question:

Should the above revised test 9.20.1.6 from recently approved 135.1-20109l-11 be accepted? Does this clarify that UNSUPPORTED_OBJECT_TYPE code is not allowed in any response from ReadProperty and ReadPropertyMultiple?

Response:

Yes. The second question is moot since the test has been changed to ensure that the ambiguous condition no longer exists in the test.