



**BACnet<sup>®</sup> TESTING LABORATORIES  
ADDENDA**

**Addendum j to  
BTL Test Package 15.2**

**Revision 7  
Revised 5/19/2019**

Approved by the BTL Working Group on 20-April-2019;  
Approved by the BTL Working Group Voting Members on 16-May-2019;  
Published on 20-May-2019.

**[This foreword and the “Overview” on the following pages are not part of this Test Package. They are merely informative and do not contain requirements necessary for conformance to the Test Package.]**

**FOREWORD**

The purpose of this addendum is to present current changes being made to the BTL Test Package. These modifications are the result of change proposals made pursuant to the continuous maintenance procedures and of deliberations within the BTL-WG Committee. The changes are summarized below.

**BTL-15.2j-1: Add COVP-B Testing [BTLWG-23][BTLWG-405][BTLWG-423] .....2**  
**BTL-15.2j-2: Add NM-FDR-A Testing [BTLWG-379] ..... 21**  
**BTL-15.2j-3: Add NM-CE-A Testing [BTLWG-400]..... 25**  
**BTL-15.2j-4: [REMOVED]..... 26**  
**BTL-15.2j-5: Update NM-RC-B Test Documentation [BTLWG-259]..... 27**

In the following document, language to be added to existing clauses within the BTL Test Package 15.2 is indicated through the use of *italics*, while deletions are indicated by ~~strike through~~. Where entirely new subclauses are proposed to be added, plain type is used throughout

In addition, changes to BTL Specified Tests also contain a **yellow** highlight to indicate the changes made by this addendum.

When this addendum is applied, all highlighting will be removed. Change markings on tests will remain to indicate the difference between the new test and an existing 135.1 test. If a test being modified has never existed in 135.1, the applied result should not contain any change markings. When this is the case, square brackets will be used to describe the changes required for this test.

Each addendum can stand independently unless specifically noted via dependency within the addendum. If multiple addenda change the same test or section, each future released addendum that changes the same test or section will note in square brackets whether or not those changes are reflected.

**BTL-15.2j-1: Add COVP-B Testing [BTLWG-23][BTLWG-405][BTLWG-423]**

**Overview:**

This document adds testing for BIBB DS-COVP-B which is a service added to BACnet in 135-1995c at Protocol\_Revision 2.

There were three work items that are combined here to result in the final version of the tests and package entries. The work items were applied in number order. i.e. BTLWG-423 super ceded the previous versions.

BTLWG-23 - introduced just the Base Requirements

BTLWG-405 - completed the testing required

BTLWG-423 - made changes to 405 (shown with strikethrough/italic and highlighted in yellow)

Additional Errata changes were made after the work items were combined to better match the COV-B section in the current test package.

**Changes:**

[In Checklist, add DS-COVP-B just after existing DS-COVP-A]

Data Sharing - Change Of Value Property - B	
R	Base Requirements
R	Supports COVP Lifetimes up to 8 hours in duration
R	Supports COVP for Status Flags changes
C <sup>1</sup>	Supports COVP for non-array property
C <sup>1</sup>	Supports COVP for array element
C <sup>1</sup>	Supports COVP for the size of an array
C <sup>1</sup>	Supports COVP for whole array
O	Supports COVP for list property
C <sup>2</sup>	Supports COVP for NULL property values
C <sup>2</sup>	Supports COVP for BOOLEAN property values
C <sup>2</sup>	Supports COVP for Enumerated property values
C <sup>2</sup>	Supports COVP for INTEGER property values
C <sup>2</sup>	Supports COVP for Unsigned property values
C <sup>2</sup>	Supports COVP for REAL property values
C <sup>2</sup>	Supports COVP for Double property values
C <sup>2</sup>	Supports COVP for Time property values
C <sup>2</sup>	Supports COVP for Date property values
C <sup>2</sup>	Supports COVP for CharacterString property values
C <sup>2</sup>	Supports COVP for OctetString property values
C <sup>2</sup>	Supports COVP for BitString property values
C <sup>2</sup>	Supports COVP for BACnetObjectIdentifier property values
C <sup>2</sup>	Supports COVP for constructed property values
C <sup>2</sup>	Supports COVP for proprietary property values of basic data types
<sup>1</sup> At least one of these options is required in order to claim conformance to this BIBB.	
<sup>2</sup> At least one of these options is required in order to claim conformance to this BIBB.	

[In BTL Test Plan, add a section for DS-COVP-B]

---

## 4.20 Data Sharing - Change Of Value Property - B

---

### 4.20.1 Base Requirements

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

BTL - 9.11.1.1 - Confirmed COV Notifications	
Test Conditionality	Must be executed.
Test Directives	Select parameters for an object and property which supports SubscribeCOVProperty. Repeat for each different datatype for which SubscribeCOVProperty is supported.
Testing Hints	

<b>BTL - 9.11.1.2 - Unconfirmed COV Notifications</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Apply the test to an object and property which supports <b>SubscribeCOVProperty</b> . Repeat for each different datatype for which <b>SubscribeCOVProperty</b> is supported..
	<b>Testing Hints</b>	
<b>135.1-2013 - 9.11.1.4 - Canceling COV Subscriptions</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.5 - Canceling Expired or Non-Existing Subscriptions</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.7 - Finite Lifetime Subscriptions</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>135.1-2013 - 9.11.1.8 - Updating Existing Subscriptions</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.9 - Client-Supplied COV Increment</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 9.11.2.1 - The Monitored Object Does Not Support COV Notification</b>		
	<b>Test Conditionality</b>	Must be executed, unless all objects support <b>SubscribeCOVProperty</b> on at least one of its properties.
	<b>Test Directives</b>	Apply the test to a property in an object that does <b>not</b> support COV (on any property).
	<b>Testing Hints</b>	
<b>BTL - 9.11.2.2 - The Monitored Property Does Not Support COV Notification</b>		
	<b>Test Conditionality</b>	Must be executed, unless all objects support <b>SubscribeCOVProperty</b> on all properties.
	<b>Test Directives</b>	Apply the test to a property for which the IUT does not support COV, which is contained in an object that does support COV (on a different property).
	<b>Testing Hints</b>	
<b>BTL - 9.11.2.X11- Monitored Object Does Not Exist</b>		
	<b>Test Conditionality</b>	Must be executed if Protocol Revision $\geq 15$
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 9.11.2.X12 - Monitored Property Does Not Exist</b>		
	<b>Test Conditionality</b>	Must be executed if Protocol Revision $\geq 15$
	<b>Test Directives</b>	Be sure to test at least one property identifier that is within the ASHRAE allocated range for standard property identifiers, but that has not yet been defined.
	<b>Testing Hints</b>	
<b>BTL - 9.11.2.X13 - There Is No Space For Subscription</b>		
	<b>Test Conditionality</b>	Must be executed
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 9.11.2.X14 - The Lifetime Parameter is Out of Range</b>		
	<b>Test Conditionality</b>	Must be executed if Protocol Revision $\geq 15$
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

### 4.20.2 Supports Lifetimes up to 8 Hours in Duration

The IUT will accept COVP subscriptions with lifetimes up to 8 hours.

<b>BTL - 9.11.1.X10 - Accepts SubscribeCOVProperty-Requests with 8 Hour Lifetimes</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	

### 4.20.3 Supports COVP for Status\_Flags Changes

The IUT supports change of value notifications for Status\_Flags changes

<b>BTL - 9.11.1.X21 - Confirmed Change of Value Notification from Status_Flags Property</b>	
<b>Test Conditionality</b>	Must be executed for each object that contains a Status_Flags and property which supports SubscribeCOVProperty. Must be executed if object type contains a Status_Flag and property which supports SubscribeCOVProperty.
<b>Test Directives</b>	Select parameters for an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty
<b>Testing Hints</b>	
<b>BTL - 9.11.1.X22 Unconfirmed Change of Value Notification from Status_Flags Property</b>	
<b>Test Conditionality</b>	Must be executed for each object that contains a Status_Flags and property which supports SubscribeCOVProperty. Must be executed if object type contains a Status_Flag and property which supports SubscribeCOVProperty.
<b>Test Directives</b>	Select parameters for an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty
<b>Testing Hints</b>	

### 4.20.4 Supports COVP to Non-array Properties

The IUT supports change of value notifications for at least one non-array property

<b>BTL - 9.11.1.1 - Confirmed COV Notifications for a SubscribeCOVProperty subscription</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Select parameters for an object and property which supports SubscribeCOVProperty..
<b>Testing Hints</b>	
<b>BTL - 9.11.1.2 - Unconfirmed COV Notifications for a SubscribeCOVProperty subscription</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Apply the test to an object and property which supports SubscribeCOVProperty. Select parameters for an object and property which supports SubscribeCOVProperty
<b>Testing Hints</b>	

### 4.20.5 Supports COVP to Array Elements

The IUT supports change of value notifications for at least one array element.

<b>BTL - 9.11.1.1 - Confirmed COV Notifications for a SubscribeCOVProperty subscription</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Select parameters for an object and property which supports SubscribeCOVProperty.
<b>Testing Hints</b>	
<b>BTL - 9.11.1.2 - Unconfirmed COV Notifications for a SubscribeCOVProperty subscription</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Apply the test to an object and property which supports SubscribeCOVProperty.

	<i>Select parameters for an object and property which supports SubscribeCOVProperty</i>
<b>Testing Hints</b>	

### 4.20.6 Supports COVP to the Size of an Array

The IUT supports change of value notifications for at least one index 0 of an array

<b>BTL - 9.11.1.1 - Confirmed COV Notifications for a SubscribeCOVProperty subscription</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Select parameters for an object and property which supports SubscribeCOVProperty</i>
<b>Testing Hints</b>	
<b>BTL - 9.11.1.2 - Unconfirmed COV Notifications for a SubscribeCOVProperty subscription</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Select parameters for an object and property which supports SubscribeCOVProperty</i>
<b>Testing Hints</b>	

### 4.20.7 Supports COVP to Whole Arrays

The IUT supports change of value notifications for at least one whole array

<b>BTL - 9.11.1.1 - Confirmed COV Notifications for a SubscribeCOVProperty subscription</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Select parameters for an object and property which supports SubscribeCOVProperty.
<b>Testing Hints</b>	
<b>BTL - 9.11.1.2 - Unconfirmed COV Notifications for a SubscribeCOVProperty subscription</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Select parameters for an object and property which supports SubscribeCOVProperty</i>
<b>Testing Hints</b>	

### 4.20.8 Supports COVP to a List Property

The IUT supports change of value notifications for at least one list property

<b>BTL - 9.11.1.1 - Confirmed COV Notifications for a SubscribeCOVProperty subscription</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Select parameters for an object and property which supports SubscribeCOVProperty.
<b>Testing Hints</b>	
<b>BTL - 9.11.1.2 - Unconfirmed COV Notifications for a SubscribeCOVProperty subscription</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Select parameters for an object and property which supports SubscribeCOVProperty</i>
<b>Testing Hints</b>	

### 4.20.9 Supports COVP to NULL Property Values

The IUT supports change of value notifications for at least one property that contains a NULL value.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty.</i>

		<i>Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

#### 4.20.10 Supports COVP to BOOLEAN Property Values

The IUT supports change of value notifications for at least one BOOLEAN property value.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

#### 4.20.11 Supports COVP to Enumerated Property Values

The IUT supports change of value notifications for at least one Enumerated property value.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

#### 4.20.12 Supports COVP to INTEGER Property Values

The IUT supports change of value notifications for at least one INTEGER property value.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty.</i>

		<i>Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

### 4.20.13 Supports COVP to Unsigned Property Values

The IUT supports change of value notifications for at least one Unsigned Property value.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

### 4.20.14 Supports COVP to REAL Property Values

The IUT supports change of value notifications for at least one REAL property value.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

### 4.20.15 Supports COVP to Double Property Values

The IUT supports change of value notifications for at least one Double property value.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty.</i>

		<i>Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

### 4.20.16 Supports COVP to Time Property Values

The IUT supports change of value notifications for at least one Time property value.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

### 4.20.17 Supports COVP to Date Property Values

The IUT supports change of value notifications for at least one Date property value.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

### 4.20.18 Supports COVP to CharacterString Property Values

The IUT supports change of value notifications for at least one CharacterString property value.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty.</i>

		<i>Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

#### 4.20.19 Supports COVP to OctetString Property Values

The IUT supports change of value notifications for at least one property with value of type OctetString.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

#### 4.20.20 Supports COVP to BitString Property Values

The IUT supports change of value notifications for at least one property with value of type BitString.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

#### 4.20.21 Supports COVP to BACnetObjectIdentifier Property Values

The IUT supports change of value notifications for at least one property with value of type BACnetObjectIdentifier.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty.</i>

		<i>Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

### 4.20.22 Supports COVP to Constructed Property Values

The IUT supports change of value notifications for at least one constructed property value.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

### 4.20.23 Supports COVP to Proprietary Property Values of Basic Data Types

The IUT supports change of value notifications for at least one proprietary property values of basic data types.

<b>BTL - 9.11.1.X11 - Confirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Select parameters for an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty.</i>
	<b>Testing Hints</b>	
<b>BTL - 9.11.1.X12 - Unconfirmed Change of Value Notification from Property Value</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	<i>Apply the test to an object and property which supports SubscribeCOVProperty. Repeat test for at least one object of each type that has at least one property which supports SubscribeCOVProperty, repeat test for all Properties which support SubscribeCOVProperty</i>
	<b>Testing Hints</b>	

[In BTL Specified Tests, derive modified versions of existing tests in 135.1-2013, for DS-COVP-B]

#### 9.11.1.1 Confirmed COV Notifications

Reason for Change: Remove the allowance for devices which do not support both confirmed and unconfirmed notifications.

Purpose: To verify that the IUT correctly responds to a SubscribeCOVProperty request to establish a subscription for confirmed COV notifications. ~~An implementation that supports COV reporting cannot respond with an error for both this test and the test in 9.11.1.2.~~

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (any valid process identifier),
  - 'Monitored Object Identifier' = (any object supporting COV notifications),
  - 'Issue Confirmed Notifications' = TRUE,
  - 'Lifetime' = (any value > 0 ~~if automatic cancellation is supported, otherwise 0~~),
  - 'Monitored Property Identifier' = (any valid property supporting COV notifications)
2. RECEIVE BACnet-SimpleACK-PDU
3. BEFORE **Notification Fail Time**
  - ~~IF (the IUT supports confirmed notifications) THEN~~
  - RECEIVE BACnetConfirmedCOVNotification-Request,
    - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
    - 'Initiating Device Identifier' = IUT,
    - 'Monitored Object Identifier' = (the same object used in the subscription),
    - 'Time Remaining' = (any value > 0 ~~if automatic cancellation is supported, otherwise 0~~),
    - 'List of Values' = (values appropriate to the property subscribed to, and any other properties the IUT provides with it, such as Status-Flags)
  - ELSE
  - RECEIVE BACnet Error PDU,
  - Error Class = SERVICES,
  - Error Code = SERVICE\_REQUEST\_DENIED|OTHER
4. TRANSMIT BACnet-SimpleACK-PDU

### 9.11.1.2 Unconfirmed COV Notifications

Reason for Change: Remove the allowance for devices which do not support both confirmed and unconfirmed notifications.

Purpose: To verify that the IUT correctly responds to a SubscribeCOVProperty request to establish a subscription for Unconfirmed COV notifications. ~~An implementation that supports COV reporting cannot respond with an error for both this test and the test in 9.11.1.1.~~

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (any valid process identifier),
  - 'Monitored Object Identifier' = (any object supporting COV notifications),
  - 'Issue Confirmed Notifications' = FALSE,
  - 'Lifetime' = (any value > 0 ~~if automatic cancellation is supported, otherwise 0~~),
  - 'Monitored Property Identifier' = (any valid property supporting COV notifications)
2. RECEIVE BACnet-SimpleACK-PDU
3. BEFORE **Notification Fail Time**
  - ~~IF (the IUT supports unconfirmed notifications) THEN~~
  - RECEIVE BACnetUnconfirmedCOVNotification-Request,
    - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
    - 'Initiating Device Identifier' = IUT,
    - 'Monitored Object Identifier' = (the same object used in the subscription),
    - 'Time Remaining' = (any value > 0 ~~if automatic cancellation is supported, otherwise 0~~),
    - 'List of Values' = (values appropriate to the property subscribed to, and any other properties the IUT provides with it, such as Status-Flags)
  - ELSE
  - RECEIVE BACnet Error PDU,
  - Error Class = SERVICES,
  - Error Code = SERVICE\_REQUEST\_DENIED|OTHER

### 9.11.1.5 Canceling Expired or Non-Existing Subscriptions

Reason for change: Added missing verification that the IUT did not send a COV notification, and removed superfluous note to tester.

Purpose: To verify that the IUT correctly responds to a SubscribeCOVProperty request to cancel a subscription that no longer exists.

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (any unused process identifier or an identifier from a previously terminated subscription),
  - 'Monitored Object Identifier' = (any unused object or an object from a previously terminated subscription),
  - 'Monitored Property Identifier' = (any unused property or a property from a previously terminated subscription)
2. RECEIVE BACnet-SimpleACK-PDU
3. **WAIT Notification Fail Time**
4. MAKE (a change to the monitored object that would cause a COV notification if there were an active subscription)
5. *CHECK(the IUT did not issue a COV notification)*

~~Notes to Tester: The IUT shall not transmit a COV notification message. An error message is not an acceptable response.~~

### 9.11.1.7 Finite Lifetime Subscriptions

Reason for change: Updates description of 'Time Remaining' and adds validation that this value counts down as expected.

Purpose: To verify that the IUT correctly responds to a SubscribeCOVProperty request to establish a subscription with a temporary lifetime. Either confirmed or unconfirmed notifications may be used, but at least one of these options must be supported by the IUT.

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (any valid process identifier),
  - 'Monitored Object Identifier' = (any object supporting COV notifications),
  - 'Issue Confirmed Notifications' = TRUE | FALSE,
  - 'Lifetime' = (a value between 60 seconds and 300 seconds),
  - 'Monitored Property Identifier' = (any valid property supporting COV notifications)
2. RECEIVE BACnet-SimpleACK-PDU
3. ~~BEFORE Notification Fail Time~~
3. IF (the subscription was for confirmed notifications) THEN
  - BEFORE Notification Fail Time*
  - RECEIVE BACnetConfirmedCOVNotification-Request,
    - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
    - 'Initiating Device Identifier' = IUT,
    - 'Monitored Object Identifier' = (the same object used in the subscription),
    - 'Time Remaining' = ~~(the requested subscription lifetime~~ *A value approximately equal to, but not greater than, the requested subscription lifetime)*,
    - 'List of Values' = (values appropriate to the property subscribed to, and any other properties the IUT provides with it, such as Status-Flags)
  - TRANSMIT BACnet-SimpleACK-PDU
- ELSE
  - BEFORE Notification Fail Time*
  - RECEIVE BACnetUnconfirmedCOVNotification-Request,
    - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
    - 'Initiating Device Identifier' = IUT,
    - 'Monitored Object Identifier' = (the same object used in the subscription),
    - 'Time Remaining'  $\sim$  (the requested lifetime)
    - 'List of Values' = (values appropriate to the property subscribed to, and any other properties the IUT provides with it, such as Status-Flags)
4. MAKE (a change to the monitored object that ~~should~~ causes a COV notification)
5. *WAIT a period longer than the resolution of the IUT's COV subscription lifetime timer*

~~5. BEFORE Notification Fail Time~~

6. IF (the subscription was for confirmed notifications) THEN

*BEFORE Notification Fail Time*

RECEIVE BACnetConfirmedCOVNotification-Request,  
 'Subscriber Process Identifier' = (the same identifier used in the subscription),  
 'Initiating Device Identifier' = IUT,  
 'Monitored Object Identifier' = (the same object used in the subscription),  
 'Time Remaining' = (TR: a value greater than 0 and less than the requested subscription lifetime),  
 'List of Values' = (values appropriate to the property subscribed to, and any other properties the IUT provides with it, such as Status-Flags)

TRANSMIT BACnet-SimpleACK-PDU

ELSE

*BEFORE Notification Fail Time*

RECEIVE BACnetUnconfirmedCOVNotification-Request,  
 'Subscriber Process Identifier' = (the same identifier used in the subscription),  
 'Initiating Device Identifier' = IUT,  
 'Monitored Object Identifier' = (the same object used in the subscription),  
 'Time Remaining' = (TR: a value greater than 0 and less than the requested subscription lifetime),  
 'List of Values' = (values appropriate to the object type of the monitored object including the changed value that triggered the notification)

7. WAIT a period longer than the resolution of the IUT's COV subscription lifetime timer

8. MAKE (a change to the monitored object that causes a COV notification)

9. IF (the subscription was for confirmed notifications) THEN

*BEFORE Notification Fail Time*

RECEIVE BACnetUnconfirmedCOVNotification-Request,  
 'Subscriber Process Identifier' = (the same identifier used in the subscription),  
 'Initiating Device Identifier' = IUT,  
 'Monitored Object Identifier' = (the same object used in the subscription),  
 'Time Remaining' = (a value greater than 0 and less than the TR),  
 'List of Values' = (values appropriate to the object type of the monitored object)

ELSE

*BEFORE Notification Fail Time*

RECEIVE BACnetUnconfirmedCOVNotification-Request,  
 'Subscriber Process Identifier' = (the same identifier used in the subscription),  
 'Initiating Device Identifier' = IUT,  
 'Monitored Object Identifier' = (the same object used in the subscription),  
 'Time Remaining' = (a value greater than 0 and less than the TR),  
 'List of Values' = (values appropriate to the object type of the monitored object including the changed value that triggered the notification)

¶10. WAIT (the lifetime of the subscription)

¶11. MAKE (a change to the monitored object that would cause a COV notification if there were an active subscription)

12. CHECK (verify that the IUT did not transmit a COV notification message)

~~Notes to Tester: The IUT shall not transmit a COV notification message addressed to the TD after step 6.~~

**9.11.1.9 Client-Supplied COV Increment**

Reason for Change: Modify the test to work with all numeric datatypes.

Purpose: To verify that the IUT correctly generates COV notifications when the client supplies the COV increment in the SubscribeCOVProperty request. Either confirmed or unconfirmed notifications may be used but at least one of these options must be supported by the IUT.

Test Concept: A subscription for COV notification is made for a property of *numeric* datatype-~~REAL~~. The subscription request specifies a COV increment. The monitored property is changed by an amount less than the increment, and the TD waits to ensure that the IUT does not generate a notification. The monitored property is changed by an amount slightly more than is required to cause a COV notification, and the TD waits for the notification.

Test Configuration: If the property being subscribed to has a related COV\_Increment property in the object, then the value of the COV\_Increment property should be significantly different than the COV increment provided in the subscription service.

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (any valid process identifier),
  - 'Monitored Object Identifier' = (any object supporting COV notifications),
  - 'Issue Confirmed Notifications' = TRUE | FALSE,
  - 'Lifetime' = (any value that will ensure no re-subscription is required to complete the test),
  - 'Monitored Property Identifier' = (any valid property supporting COV notifications),
  - 'COV Increment' = (any valid increment value)
2. RECEIVE BACnet-SimpleACK-PDU
3. BEFORE **Notification Fail Time**
  - IF (the subscription was for confirmed notifications) THEN
    - RECEIVE BACnetConfirmedCOVNotification-Request,
      - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
      - 'Initiating Device Identifier' = IUT,
      - 'Monitored Object Identifier' = (the same object used in the subscription),
      - 'Time Remaining' ~ = (the requested lifetime),
      - 'List of Values' = (values appropriate to the object type of the monitored object including the value of monitored property)
    - TRANSMIT BACnet-SimpleACK-PDU
  - ELSE
    - RECEIVE BACnetUnconfirmedCOVNotification-Request,
      - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
      - 'Initiating Device Identifier' = IUT,
      - 'Monitored Object Identifier' = (the same object used in the subscription),
      - 'Time Remaining' ~ = (the requested lifetime),
      - 'List of Values' = (values appropriate to the object type of the monitored object including the value of monitored property)
4. MAKE (the monitored property change by less than the COV increment)
5. CHECK (verify that the IUT did not transmit a notification message for the monitored property)
6. MAKE (the monitored property change by slightly more than COV Increment less the amount changed in step 5.4)
7. BEFORE **Notification Fail Time**
  - IF (the subscription was for confirmed notifications) THEN
    - RECEIVE BACnetConfirmedCOVNotification-Request,
      - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
      - 'Initiating Device Identifier' = IUT,
      - 'Monitored Object Identifier' = (the same object used in the subscription),
      - 'Time Remaining' = ?,
      - 'List of Values' = (values appropriate to the object type of the monitored object including the changed value that triggered the notification)
    - TRANSMIT BACnet-SimpleACK-PDU
  - ELSE
    - RECEIVE BACnetUnconfirmedCOVNotification-Request,
      - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
      - 'Initiating Device Identifier' = IUT,
      - 'Monitored Object Identifier' = (the same object used in the subscription),
      - 'Time Remaining' = ?,
      - 'List of Values' = (values appropriate to the object type of the monitored object including the changed value that triggered the notification)

[In BTL Specified Tests, derive modified versions of two existing tests in 135.1-2013, with specified responses different from those in the modified versions of those tests in 135.1-2013o, for DS-COV-B]

### 9.11.2.1 The Monitored Object Does Not Support COV Notification

*Reason for Change: Update the accepted error responses as per changes made in Protocol Revision 15.*

Purpose: To verify that the IUT correctly responds to a SubscribeCOVProperty request to establish a subscription when the monitored object does not support COV notifications.

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (any valid process identifier),
  - 'Monitored Object Identifier' = (any object that does not support COV notifications),
  - 'Issue Confirmed Notifications' = TRUE,
  - 'Lifetime' = 60,
  - 'Monitored Property Identifier' = (any property in the object)
- ~~2. RECEIVE BACnet-Error-PDU,~~
  - ~~Error Class = SERVICES,~~
  - ~~Error Code = SERVICE\_REQUEST\_DENIED | OTHER~~
2. IF (Protocol\_Revision < 15) THEN
  - RECEIVE
    - (BACnet-Error-PDU,
    - Error Class = SERVICES,
    - Error Code = SERVICE\_REQUEST\_DENIED | OTHER) |
    - (BACnet-Error-PDU,
    - Error Class = OBJECT,
    - Error Code = OPTIONAL\_FUNCTIONALITY\_NOT\_SUPPORTED) |
    - (BACnet-Error-PDU,
    - Error Class = PROPERTY,
    - Error Code = NOT\_COV\_PROPERTY)
  - ELSE
    - RECEIVE
      - (BACnet-Error-PDU,
      - Error Class = OBJECT,
      - Error Code = OPTIONAL\_FUNCTIONALITY\_NOT\_SUPPORTED) |
      - (BACnet-Error-PDU,
      - Error Class = PROPERTY,
      - Error Code = NOT\_COV\_PROPERTY)

### 9.11.2.2 The Monitored Property Does Not Support COV Notification

*Reason for Change: Update the accepted error responses as per changes made in Protocol\_Revision 15.*

Purpose: To verify that the IUT correctly responds to a SubscribeCOVProperty request to establish a subscription when the monitored object supports COV notifications but not on the requested property.

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (any valid process identifier),
  - 'Monitored Object Identifier' = (any object that supports COV notifications),
  - 'Issue Confirmed Notifications' = TRUE,
  - 'Lifetime' = 60,
  - 'Monitored Property Identifier' = (any property of the chosen object that does not support COV notifications)
- ~~2. RECEIVE BACnet-Error-PDU,~~
  - ~~Error Class = SERVICES,~~
  - ~~Error Code = SERVICE\_REQUEST\_DENIED | OTHER~~
2. IF (Protocol\_Revision < 15) THEN
  - RECEIVE
    - (BACnet-Error-PDU,
    - Error Class = SERVICES,
    - Error Code = SERVICE\_REQUEST\_DENIED | OTHER) |
    - (BACnet-Error-PDU,
    - Error Class = PROPERTY,
    - Error Code = NOT\_COV\_PROPERTY)
  - ELSE
    - RECEIVE BACnet-Error-PDU,
    - Error Class = PROPERTY,

*Error Code = NOT\_COV\_PROPERTY*

[In BTL Specified Tests, add new tests as shown]

**9.11.2.X11 Monitored Object Does Not Exist**

Purpose: To verify that the IUT correctly responds to a SubscribeCOVProperty request to establish a subscription when the monitored object does not exist.

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (any valid process identifier),
  - 'Monitored Object Identifier' = (any object of a type that supports COV and an instance which does not exist in the IUT),
  - 'Issue Confirmed Notifications' = TRUE,
  - 'Lifetime' = 60
  - 'Monitored Property Identifier' = (any valid property supporting COV notifications)
2. RECEIVE BACnet-Error-PDU,
  - Error Class = OBJECT,
  - Error Code = UNKNOWN\_OBJECT

**9.11.2.X12 Monitored Property Does Not Exist**

Purpose: To verify that the IUT correctly responds to a SubscribeCOVProperty request to establish a subscription when the monitored property does not exist.

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (any valid process identifier),
  - 'Monitored Object Identifier' = (object supporting COV notifications),
  - 'Issue Confirmed Notifications' = TRUE,
  - 'Lifetime' = 60
  - 'Monitored Property Identifier' = (any valid property supporting COV notifications which does not exist for specified object)
2. RECEIVE BACnet-Error-PDU,
  - Error Class = PROPERTY,
  - Error Code = UNKNOWN\_PROPERTY

**9.11.2.X13 There Is No Space For Subscription**

Purpose: To verify that the IUT correctly responds to a SubscribeCOVProperty request to establish a subscription when there is no space for a subscription.

Test Concept: Repeatedly subscribe to the same object each time with a different Process Identifier until the device runs out of resources and returns the appropriate error. This test only applies to IUTs that claim a Protocol\_Revision of 10 or higher.

Test Conditionality: If the device cannot be configured such that the maximum number of subscriptions the IUT can accept is less than 10000, then this test may be skipped.

Test Steps:

REPEAT PID = (1 through the maximum number of subscriptions the IUT can accept plus 1, or until the IUT returns an Error-PDU) {

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = PID,
  - 'Monitored Object Identifier' = (object supporting COV notifications),
  - 'Issue Confirmed Notifications' = TRUE,
  - 'Lifetime' = 6000
  - 'Monitored Property Identifier' = (any valid property supporting COV notifications)
2. **RECEIVE BACnet-SimpleACK-PDU** (BACnet-Error-PDU,
  - Error Class = RESOURCES,

```

Error Code = NO_SPACE_TO_ADD_LIST_ELEMENT)
3. READ ACS = (Active_COV_Subscriptions)
4. IF (a BACnet Simple Ack was received in step 2) THEN
    CHECK (that the subscription is in ACS)
ELSE
    CHECK (that the subscription is not in ACS)
}

```

### 9.11.2.X14 The Lifetime Parameter is Out of Range

Purpose: To verify that the IUT correctly responds to a SubscribeCOVProperty request to establish a subscription when the Lifetime parameter is out of range.

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (any valid process identifier),
  - 'Monitored Object Identifier' = (object supporting COV notifications),
  - 'Issue Confirmed Notifications' = TRUE,
  - 'Lifetime' = (a value larger than that supported by the IUT),
  - 'Monitored Property Identifier' = (any valid property supporting COV notifications)
2. IF (Protocol\_Revision is present and Protocol\_Revision => 15) THEN
  - RECEIVE BACnet-Error-PDU,
    - Error Class = SERVICES,
    - Error Code = VALUE\_OUT\_OF\_RANGE
  - ELSE
    - RECEIVE BACnet-Error-PDU,
      - Error Class = SERVICES,
      - Error Code = VALUE\_OUT\_OF\_RANGE | SERVICE\_REQUEST\_DENIED | OTHER
        - | (RECEIVE BACnet-Reject-PDU,
        - Reject Reason = PARAMETER\_OUT\_OF\_RANGE)

### 9.11.1.X10 Accepts SubscribeCOVProperty-Requests with 8 Hour Lifetimes

Purpose: To verify that the IUT correctly accepts lifetimes of at least 8 hours.

Test Steps:

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (any valid process identifier),
  - 'Monitored Object Identifier' = (any object supporting COV notifications),
  - 'Issue Confirmed Notifications' = TRUE | FALSE,
  - 'Lifetime' = 28800
  - 'Monitored Property Identifier' = (any valid property supporting COV notifications)
2. RECEIVE BACnet-SimpleACK-PDU
3. BEFORE Notification Fail Time
  - IF (the subscription was for confirmed notifications) THEN
    - RECEIVE BACnetConfirmedCOVNotification-Request,
      - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
      - 'Initiating Device Identifier' = IUT,
      - 'Monitored Object Identifier' = (the same object used in the subscription),
      - 'Time Remaining' ~= (the requested lifetime),
      - 'List of Values' = (values appropriate to the object type of the monitored object including the value of monitored property)
    - TRANSMIT BACnet-SimpleACK-PDU
  - ELSE
    - RECEIVE BACnetUnconfirmedCOVNotification-Request,
      - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
      - 'Initiating Device Identifier' = IUT,
      - 'Monitored Object Identifier' = (the same object used in the subscription),

'Time Remaining' = (the requested lifetime),  
 'List of Values' = (values appropriate to the object type of the monitored object including the value of monitored property)

4. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (the same identifier used in Step 1),
  - 'Monitored Object Identifier' = (the same identifier used in the subscription),
  - 'Monitored Property Identifier' = (the same object used in the subscription)

**5. RECEIVE BACnet-SimpleACK-PDU**

**9.11.1.X11 Confirmed Change of Value Notification from Property Value**

Purpose: To verify that the IUT can initiate ConfirmedCOVNotification service requests conveying a change of the Property Value.

Test Concept: A property subscription for COV notifications is established, using a Lifetime of L. L shall be set to a value less than 24 hours and large enough to complete the test. The Value of the monitored Property is changed and a notification shall be received. The subscribed property may be changed using the WriteProperty service or by another means. For implementations where it is not possible to write to these properties at all the vendor shall provide an alternative trigger mechanism to accomplish this task. All of these methods are equally acceptable.

Test Steps:

~~REPEAT X = (one object of each type that object and property which supports SubscribeCOVProperty) DO {~~  
~~REPEAT Y = (each Property which supports SubscribeCOVProperty) DO {~~

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (any valid process identifier),
  - 'Monitored Object Identifier' = X
  - 'Issue Confirmed Notifications' = TRUE,
  - 'Lifetime' = L
  - 'Monitored Property Identifier' = Y (any valid property supporting COV notifications)
2. RECEIVE BACnet-SimpleACK-PDU
3. BEFORE Notification Fail Time
  - RECEIVE BACnetConfirmedCOVNotification-Request,
    - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
    - 'Initiating Device Identifier' = IUT,
    - 'Monitored Object Identifier' = X
    - 'Time Remaining' = (any value appropriate for the Lifetime selected),
    - 'List of Values' = (values appropriate to the property subscribed to, and any other properties the IUT provides

with it, such as Status-Flags)

4. TRANSMIT BACnet-SimpleACK-PDU
5. MAKE (a change to the monitored object PROPERTY that causes a COV notification)
6. BEFORE Notification Fail Time

- RECEIVE BACnetConfirmedCOVNotification-Request,
  - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
  - 'Initiating Device Identifier' = IUT,
  - 'Monitored Object Identifier' = X
  - 'Time Remaining' = (any value appropriate for the Lifetime selected),
  - 'List of Values' = (values appropriate to the property subscribed to, and any other properties the IUT provides

with it, such as Status-Flags)

7. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (the same identifier used in Step 1),
  - 'Monitored Object Identifier' = X
  - 'Monitored Property Identifier' = Y

**8. RECEIVE BACnet-SimpleACK-PDU**

→  
 }

**9.11.1.X12 Unconfirmed Change of Value Notification from Property Value**

Purpose: To verify that the IUT can initiate UnconfirmedCOVNotification service requests conveying a change of the Property Value.

Test Steps: The steps for this test case are identical to the test steps in 9.11.1.X11 except that the SubscribeCOVProperty service request in step 1 shall have a value of FALSE for the 'Issue Confirmed Notifications' parameter, all of the ConfirmedCOVNotification requests shall be UnconfirmedCOVNotification requests, and there is no acknowledgment of the unconfirmed services. The MAC address used for the notification message shall be such that the TD is one of the recipients

### 9.11.1.X21 Confirmed Change of Value Notification from Status\_Flags Property

Purpose: To verify that the IUT can initiate ConfirmedCOVNotification service requests conveying a change of the Status\_Flags Property.

Test Concept: A property subscription for COV notifications is established, using a Lifetime of L. L shall be set to a value less than 24 hours and large enough to complete the test. . The Status\_Flags property of the monitored object is then changed and a notification shall be received. The value of the Status-Flags property can be changed by using the WriteProperty service or by another means. For some implementations writing to the Out\_Of\_Service property will accomplish this task. For implementations where it is not possible to write to Status\_Flags or Out\_Of\_Service or change the Status\_Flags by any other means, this test shall be skipped.

Test Steps:

~~REPEAT X = (one object of each type that object and property which supports SubscribeCOVProperty) DO {~~  
~~REPEAT Y = (each Property which supports SubscribeCOVProperty) DO {~~

1. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (any valid process identifier),
  - 'Monitored Object Identifier' = X
  - 'Issue Confirmed Notifications' = TRUE,
  - 'Lifetime' = L
  - 'Monitored Property Identifier' = Y (any valid property supporting COV notifications)
2. RECEIVE BACnet-SimpleACK-PDU
3. BEFORE Notification Fail Time
  - RECEIVE BACnetConfirmedCOVNotification-Request,
    - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
    - 'Initiating Device Identifier' = IUT,
    - 'Monitored Object Identifier' = X
    - 'Time Remaining' = (any value appropriate for the Lifetime selected),
    - 'List of Values' = (values appropriate to the property subscribed to and initial Status\_Flags)
4. TRANSMIT BACnet-SimpleACK-PDU
5. MAKE (Status\_Flags = any value that differs from "initial Status\_Flags")
6. BEFORE Notification Fail Time
  - RECEIVE BACnetConfirmedCOVNotification-Request,
    - 'Subscriber Process Identifier' = (the same identifier used in the subscription),
    - 'Initiating Device Identifier' = IUT,
    - 'Monitored Object Identifier' = X
    - 'Time Remaining' = (any value appropriate for the Lifetime selected),
    - 'List of Values' = (initial values appropriate to the property subscribed to and new Status\_Flags)
7. TRANSMIT SubscribeCOVProperty-Request,
  - 'Subscriber Process Identifier' = (the same identifier used in Step 1),
  - 'Monitored Object Identifier' = X
  - 'Monitored Property Identifier' = Y

8. RECEIVE BACnet-SimpleACK-PDU

— }  
 }

### 9.11.1.X22 Unconfirmed Change of Value Notification from Status\_Flags Property

Purpose: To verify that the IUT can initiate UnconfirmedCOVNotification service requests conveying a change of the Status\_Flags Property.

Test Steps: The steps for this test case are identical to the test steps in 9.11.1.X21 except that the SubscribeCOVProperty service request in step 1 shall have a value of FALSE for the 'Issue Confirmed Notifications' parameter, all of the ConfirmedCOVNotification requests shall be UnconfirmedCOVNotification requests, and there is no acknowledgment of the unconfirmed services. The MAC address used for the notification message shall be such that the TD is one of the recipients

**BTL-15.2j-2: Add NM-FDR-A Testing [BTLWG-379]**

**Overview:**

Addendum 135-2012*al* added the NM-FDR-A BIBB. This document makes needed changes in the BTL Test Package to provide for claiming the BIBB.

These changes are not contained in any SSPC proposal.

**Changes:**

[In BTL Checklist, add new Network Management - Foreign Device Registration - A section]

**10 Network Management**

Support	Listing	Option
<b>Network Management - Foreign Device Registration - A</b>		
	<b>R</b>	Base Requirements
	<b>BTL-R</b>	Supports configurable BBMD Address
	<b>O</b>	Supports a mode where it transmits a Broadcast at Startup
	<b>O</b>	Supports configurable Time-to-Live

[In BTL Test Plan, add new Network Management - Foreign Device Registration -A sections at end of section 10]

**10 Network Management**

**10.X2 Network Management - Foreign Device Registration - A**

These tests are designed for testing the recurring initiation of a Register-Foreign-Device BVLL to the configured BBMD.

**10.X2.1 Base Requirements**

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

<b>135.1-2013 - 14.8 - Register-Foreign-Device Test</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 14.9.X1 - Register-Foreign-Device Enable and Disable Test</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 14.9.X2 - Recurring Register-Foreign-Device Test</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>135.1-2013 - 14.1.6 - Distribute-Broadcast-To-Network</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>135.1-2013 - 14.1.9 - Original-Unicast-NPDU</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 14.1.10 - Forwarded-NPDU (Two-hop Distribution)</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

## 10.X2.2 Supports Configurable BBMD Address

The IUT supports a configurable BBMD Address to which it sends Register-Foreign-Device NPDUs.

<b>BTL - 14.9.X3 - BBMD Address Configuration Test</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	

## 10.X2.3 Supports a Mode Where it Transmits a Broadcast at Startup

The IUT transmits a Broadcast at Startup, which can be observed preceded by the sending of Register-Foreign-Device NPDUs, when configured as a Foreign Device.

<b>BTL - 14.9.X4 - Transmits a Broadcast at Startup preceded by Register-Foreign-Device</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	

## 10.X2.4 Supports Configurable Time-to-Live

The IUT supports a configurable Time-to-Live which it uses in the Register-Foreign-Device NPDUs it sends.

<b>BTL - 14.9.X5 - Time-to-Live Configuration Test</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	

[Add in BTL Specified tests, these five entirely new tests]

### 14.9.X1 Register-Foreign-Device Enable and Disable Test

Reason For Change: This tests that the behavior in test 14.8 can be configured by the product end-user.

Purpose: Verify that the option to issue Register-Foreign-Device requests can be configured by the product end-user.

Test Concept: Using a product end-user interface, configure the mode for use of Register-Foreign-Device requests, and then configure the mode to cease use of Register-Foreign-Device requests.

Configuration Requirements: The means by which the product is here configured shall be part of the product's end-user interface. BBMD1 is the TD simulating a correctly functioning BBMD implementation.

Test Steps:

1. MAKE (IUT enter mode for use of Register-Foreign-Device requests)
2. RECEIVE DA = BBMD1,  
Register-Foreign-Device
3. TRANSMIT BVLC-Result,  
'Result Code' = Successful completion
4. MAKE (the IUT not in mode for use of Register-Foreign-Device requests)
5. WAIT (more than 31 seconds longer than the 'Time-to-Live' parameter used in Register-Foreign-Device requests)
6. CHECK (that the IUT did not send any Register-Foreign-Device requests)

### 14.9.X2 Recurring Register-Foreign-Device Test

Reason For Change: This tests in continuous manner what 14.9.1 tests just once.

Purpose: Verify that mode for use of Register-Foreign-Device and setting of 'BBMD Address' parameter are persistent across reset, and that the issuance of Register-Foreign-Device precedes the first issuance of any broadcast, when in that mode.

Test Concept: IUT is put in a mode to use Register-Foreign-Device requests, and it is observed that Register-Foreign-Device requests are sent sufficiently frequently to prevent expiration of the registration at the BBMD.

Configuration Requirements: The product's setting of 'BBMD Address' parameter is configured as BBMD1. BBMD1 is the TD simulating a correctly functioning BBMD implementation.

Test Steps:

1. MAKE (IUT enter mode for use of Register-Foreign-Device requests)

2. RECEIVE DA = BBMD1,  
Register-Foreign-Device
3. TRANSMIT BVLC-Result,  
'Result Code' = Successful completion
4. BEFORE (the time configured for the 'Time-to-Live' parameter used for Register-Foreign-Device requests)  
RECEIVE DA = BBMD1,  
Register-Foreign-Device
5. TRANSMIT BVLC-Result,  
'Result Code' = Successful completion
6. BEFORE (the time configured for the 'Time-to-Live' parameter used for Register-Foreign-Device requests)  
RECEIVE DA = BBMD1,  
Register-Foreign-Device
7. TRANSMIT BVLC-Result,  
'Result Code' = Successful completion

Notes to Tester: There is no need for the recurring request to be sent any more quickly than precisely the 'Time-to-Live' since the standard mandates that the BBMD preserve the registration for 30 seconds past the 'Time-to-Live'.

#### **14.9.X3 BBMD Address Configuration Test**

Reason For Change: This tests that the behavior in test 14.8 can be configured by the product end-user.

Purpose: Verify that the parameter in Register-Foreign-Device in test 14.8 can be configured by the product end-user.

Test Concept: Using a product end-user interface, configure the 'BBMD Address' parameter that is used in Register-Foreign-Device requests.

Configuration Requirements: The means by which the product is configured for a 'BBMD Address' can be anything in the product's end-user interface. BBMD1 is the TD simulating a correctly functioning BBMD implementation.

Test Steps:

1. MAKE (through the product's end-user interface, the setting of 'BBMD Address' parameter equal BBMD1)
2. MAKE (IUT enter mode for use of Register-Foreign-Device requests)
3. RECEIVE DA = BBMD1,  
Register-Foreign-Device
4. TRANSMIT BVLC-Result,  
'Result Code' = Successful completion

#### **14.9.X4 Transmits a Broadcast at Startup preceded by Register-Foreign-Device**

Reason For Change: This tests in the specific case of startup, what test 14.9.1 expects to observe during ordinary ongoing operation.

Purpose: Verify that mode for use of Register-Foreign-Device and setting of 'BBMD Address' parameter are persistent across reset, and that the issuance of Register-Foreign-Device precedes the first issuance of any broadcast, when in that mode.

Test Concept: IUT is put in a mode to use Register-Foreign-Device requests, persistently so it will be re-established, then IUT is reset, and the timing of Register-Foreign-Device request to re-establish that precedes the first issuance of any broadcast.

Configuration Requirements: The product's setting of 'BBMD Address' parameter is configured as BBMD1. BBMD1 is the TD simulating a correctly functioning BBMD implementation.

Test Steps:

1. MAKE (IUT enter mode for use of Register-Foreign-Device requests, persistently  
so it will be re-established after any reset)
2. MAKE (IUT reset)
3. RECEIVE DA = BBMD1,  
Register-Foreign-Device
4. TRANSMIT BVLC-Result,  
'Result Code' = Successful completion
5. RECEIVE DA = BBMD1,  
Distribute-Broadcast-To-Network,  
NPDU = (any broadcast)
6. TRANSMIT BVLC-Result,

'Result Code' = Successful completion

Notes to Tester: For the I-Am, one can precede the Register-Foreign-Device command, as long as then after the Register-Foreign-Device occurs, it is followed by a Distribute-Broadcast-To-Network again, of that I-Am.

#### **14.9.X5 Time-to-Live Configuration Test**

Reason For Change: Adds verification that the behavior in test 14.8 can be configured by the product end-user.

Purpose: Verify that the parameter in Register-Foreign-Device in test 14.8 can be configured by the product end-user, through a reasonable range (120 through 28800 is sufficient; the absolute upper limit is 65535 seconds, approximately 17 hours).

Test Concept: Using a product end-user interface, configure the 'Time-to-Live' parameter that is used in Register-Foreign-Device requests.

Configuration Requirements: The means by which the product is configured can be anything in the product's end-user interface. BBMD1 is the TD simulating a correctly functioning BBMD implementation.

Test Steps:

1. MAKE (through the product's end-user interface, the setting of 'Time-to-Live' parameter equal 120, or any larger value supported by the implementation)
2. MAKE (IUT enter mode for use of Register-Foreign-Device requests)
3. RECEIVE DA = BBMD1,  
Register-Foreign-Device,  
'Time-to-Live' = (value configured in step 1)
4. TRANSMIT BVLC-Result,  
'Result Code' = Successful completion

**BTL-15.2j-3: Add NM-CE-A Testing [BTLWG-400]**

**Overview:**

Add testing for NM-CE-A.

**Changes:**

[In BTL Checklist, modify Network Management - Connection Establishment - A as shown below]

<b>Network Management - Connection Establishment - A</b>		
	R <sup>+</sup>	Base Requirements
†Contact BTL for interim tests for this BIBB.		

[In BTL Test Plan, append to Section 10, Network Management]

---

## **10.X4 Network Management - Connection Establishment - A**

---

### **10.X4.1 Base Requirements**

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

<b>135.1-2013 - 10.5.3.1 - Establish-Connection-To-Network</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>135.1-2013 - 10.5.3.2 - Disconnect-Connection-To-Network</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

**BTL-15.2j-4: [REMOVED]**

## **BTL-15.2j-5: Update NM-RC-B Test Documentation [BTLWG-259]**

### **Overview:**

All testing which should be applied is best expressed in the sections for physical or virtual routing. All the irrelevant and outdated discussion of three variants, and of half-routing, in the existing first section of Network Management should be removed.

Typographic convention: In this proposal, elisions are shown with ~~strike-through~~ and additions are shown in *italics*.

### **Changes:**

[In Test Plan, make these changes in the existing first section, as shown.]

## **10.1 Network Management - Routing**

### **10.1.1 Base Requirements**

~~There are three classes of routing functionality: A BACnet Gateway, which connects a physical BACnet LAN and one or more virtual BACnet LANs (per clause H.1 and H.2 in ASHRAE 135).~~

~~We envision three classes of routing functionality:~~

~~Routing between physical BACnet LANs~~

~~Half routing (as per clause 6.7 in ASHRAE 135)~~

~~Routing between a physical BACnet LAN and a virtual BACnet LAN (as per clauses H.1 and H.2 in ASHRAE 135)~~

~~A device may support one or more of these classes.~~

There are no base requirements tests for this section.