

**[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-TP14.0d-1: Add Support for Global Group Object, pg 2.** [wID0047]

In the following document, language to be added to existing clauses within the BTL Test Package 14.0 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 might 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.

This addendum contains results of various clarification requests put forth to the BTL-WG that resulted in test package changes.

**BTL-TP14.0d-1: Add Support for Global Group Object**

**Overview:**

This document applies changes to the Checklist, Test Plan and Specified Tests for:

- Add Global Group object type.

**Changes:**

[In BTL Checklist, add Global Group object type to Section 3, Objects]

Support	Listing	Option
<b>Global Group Object</b>		
	R	Base Requirements
	S	Supports writable Out Of Service properties
	O	Supports writable Group Members properties
	BTL-C <sup>1</sup>	Supports DS-RP-A for retrieving member values
	O <sup>2</sup>	Is able to read REAL values
	O <sup>2,3</sup>	Is able to read Unsigned values
	O <sup>2,3</sup>	Is able to read INTEGER (Signed) values
	O <sup>2</sup>	Is able to read BOOLEAN values
	O <sup>2</sup>	Is able to read Bit String values
	O <sup>2,4</sup>	Is able to read Enumerated values
	O <sup>2</sup>	Is able to read NULL values
	O	Is able to read Double values
	O	Is able to read Character String values
	O	Is able to read Octet String values
	O	Is able to read Date values
	O	Is able to read Time values
	O	Is able to read BACnetObjectIdentifier values
	O	Supports COV to retrieve remote property values
	O <sup>5,6</sup>	Supports COVU Period and COVU Recipients properties
	O	Supports Reliability Evaluation
<p><sup>1</sup> If the Group_Members property is writable, DS-RP-A shall be supported.</p> <p><sup>2</sup> If the Group_Members property is writable, this data type must be supported.</p> <p><sup>3</sup> Reading of 32 bit Unsigned and INTEGER values is minimally required.</p> <p><sup>4</sup> Reading of 16-bit Enumerated values is minimally required.</p> <p><sup>5</sup> If COVU_Period or COVU_Recipients properties are present then both shall be present and writable.</p> <p><sup>6</sup> If COVU_Period or COVU_Recipients properties are present then DM-DDB-A and DS-COVU-B shall be supported.</p>		

[In BTL Checklist, add CHANGE OF STATUS FLAGS to Section 5, Alarm and Event Management BIBBs]

Support	Listing	Option
<b>Alarm and Event - Notification - Internal - B</b>		
	R	Base Requirements
	R	Supports AE-INFO-B
	R	Supports the Notification Class Object
	C <sup>1</sup>	Supports AE-ACK-B
	C <sup>2</sup>	Implements intrinsic alarming
	C <sup>2</sup>	Supports the Event Enrollment object

Support	Listing	Option
	C <sup>3</sup>	Implements the CHANGE OF BITSTRING algorithm
	C <sup>3</sup>	Implements the CHANGE OF STATE algorithm
	C <sup>3</sup>	Implements the numeric form of the CHANGE OF VALUE algorithm
	C <sup>3</sup>	Implements the Bit String form of the CHANGE OF VALUE algorithm
	C <sup>3</sup>	Implements the COMMAND FAILURE algorithm
	C <sup>3</sup>	Implements the FLOATING LIMIT algorithm
	C <sup>3</sup>	Implements the OUT OF RANGE algorithm
	C <sup>3</sup>	Implements a Proprietary algorithm using Complex notifications
	C <sup>3</sup>	Implements a Proprietary algorithm using Extended notifications
	C <sup>4</sup>	Generates event notifications with timestamps of the BACnetDateTime form
	C <sup>4</sup>	Generates event notifications with timestamps of the Time form
	C <sup>4</sup>	Generates event notifications with timestamps of the Sequence Number form
	O	Implements intrinsic alarming in an Analog object
	O	Supports writable Event Parameters properties
	C <sup>3</sup>	Implements the DOUBLE OUT OF RANGE Algorithm
	C <sup>3</sup>	Implements the SIGNED OUT OF RANGE Algorithm
	C <sup>3</sup>	Implements the UNSIGNED OUT OF RANGE Algorithm
	C <sup>3</sup>	Implements the CHANGE OF CHARACTERSTRING Algorithm
	C <sup>3</sup>	Implements the CHANGE OF STATUS FLAGS algorithm
	O	Supports Event Message Texts property
	O	Supports Event Message Texts Config property
<p><sup>1</sup> Required if EventNotifications with service parameter AckRequired = True can be issued.  <sup>2</sup> At least one of these options must be supported to claim support for this BIBB.  <sup>3</sup> At least one of these options must be supported to claim support for this BIBB. It is recommended that a standard BACnet algorithm be used instead of a proprietary algorithm whenever possible.  <sup>4</sup> At least one of these options must be supported to claim support for this BIBB. The BACnetDateTime form of the timestamp is the recommended option.</p>		
<b>Alarm and Event - Notification - External - B</b>		
	R	Base Requirements
	R	Supports AE-N-I-B
	R	Supports DS-RP-A for retrieving monitored values
	R	Supports the Event Enrollment object
	C <sup>1</sup>	Implements the CHANGE OF BITSTRING algorithm
	C <sup>1</sup>	Implements the CHANGE OF STATE algorithm
	C <sup>1</sup>	Implements the numeric form of the CHANGE OF VALUE algorithm
	C <sup>1</sup>	Implements the Bit String form of the CHANGE OF VALUE algorithm
	C <sup>1</sup>	Implements the COMMAND FAILURE algorithm
	C <sup>1</sup>	Implements the FLOATING LIMIT algorithm
	C <sup>1</sup>	Implements the OUT OF RANGE algorithm
	C <sup>1</sup>	Implements the DOUBLE OUT OF RANGE algorithm
	C <sup>1</sup>	Implements the SIGNED OUT OF RANGE algorithm
	C <sup>1</sup>	Implements the UNSIGNED OUT OF RANGE algorithm
	C <sup>1</sup>	Implements the CHANGE OF CHARACTERSTRING algorithm
	C <sup>1</sup>	Implements the CHANGE OF STATUS FLAGS algorithm
	C <sup>1</sup>	Implements a Proprietary algorithm
	O	Supports Event Message Texts property.
<p><sup>1</sup> One of these options must be supported to claim support for this BIBB. It is recommended that a standard BACnet algorithm be used instead of a proprietary algorithm whenever possible.</p>		

[In BTL Test Plan, add Global Group object tests in section 3.X36]

### 3.X36 Global Group Object

#### 3.X36.1 Base Requirements

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

<b>135.1-2013 - 7.2.3 – Read-only Property Test</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>ASHRAE 135.1-2011</i>
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Test the Present Value property of each Global Group object.
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	
<b>BTL - 7.3.2.13.X2 - Reliability MEMBER_FAULT Test</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	If no object pointed to by the Group_Members property can be made to contain Status_Flags FAULT flag equal to TRUE, then this test shall be skipped.
<b>Test Directives</b>	
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	
<b>BTL - 7.3.2.13.X3 - Reliability COMMUNICATION_FAILURE Test</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	If the Groups_Members property is not writable and can not be made to contain references to external objects, this test shall be skipped.
<b>Test Directives</b>	Repeat this test with a member of the Group_Members property pointing to a device not communicating and with a member of the Group_Members property pointing to a device responding with a BACnet-Error-PDU.
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	
<b>BTL - 7.3.2.13.X4 - Present Value Tracking and Reliability Test</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	If the Reliability property can not be made to not equal NO_FAULT_DETECTED, this test shall be omitted.
<b>Test Directives</b>	The test shall be executed using a Global Group object.
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	

#### 3.X36.2 Supports Writable Out\_Of\_Service Properties

The Out\_Of\_Service property in the Global Group object contained in the IUT are writable.

<b>BTL - 7.3.2.13.X1 - Global Group Present Value, Out Of Service and Status Flags Test</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	If Out Of Service can be made TRUE, this test must be executed.
<b>Test Directives</b>	

	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	

### 3.X36.3 Supports writable Group\_Members properties

The property in the Global Group object contained in the IUT is writable.

<b>135.1-2013 - 7.3.2.13.1 - Resizing Group_Member_Names by Writing Group_Members Property Test</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>ASHRAE 135.1-2013</i> .
	<b>Test Conditionality</b>	If this property is writable, this test must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	
<b>135.1-2013 - 7.3.2.13.2 - Resizing Group_Members by Writing Group_Member_Names Property Test</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>ASHRAE 135.1-2013</i> .
	<b>Test Conditionality</b>	If this property is writable, this test must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	

### 3.X36.4 Is able to read REAL values

The IUT supports a Global Group object that is able to read REAL values.

<b>BTL - 7.3.2.13.X5 - Present Value Tracking Test</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Set the Group_Members property to make the Present_Value read a REAL value.
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	

### 3.X36.5 Is able to read Unsigned values

The IUT supports a Global Group object that is able to read Unsigned values.

<b>BTL - 7.3.2.13.X5 - Present Value Tracking Test</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Set the Group_Members property to make the Present_Value read an UNSIGNED value.
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	

### 3.X36.6 Is able to read INTEGER (Signed) values

The IUT supports a Global Group object that is able to read INTEGER values.

BTL - 7.3.2.13.X5 - Present Value Tracking Test		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Set the Group_Members property to make the Present_Value read an INTEGER (Signed) value.
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	

### 3.X36.7 Is able to read BOOLEAN values

The IUT supports a Global Group object that is able to read BOOLEAN values.

BTL - 7.3.2.13.X5 - Present Value Tracking Test		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Set the Group_Members property to make the Present_Value read a BOOLEAN value.
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	

### 3.X36.8 Is able to read Bit String values

The IUT supports a Global Group object that is able to read Bit String values.

BTL - 7.3.2.13.X5 - Present Value Tracking Test		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Set the Group_Members property to make the Present_Value read a Bit String value.
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	

### 3.X36.9 Is able to read Enumerated values

The IUT supports a Global Group object that is able to read Enumerated values.

BTL - 7.3.2.13.X5 - Present Value Tracking Test		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Set the Group_Members property to make the Present_Value read an Enumerated value.
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	

### 3.X36.10 Is able to read NULL values

The IUT supports a Global Group object that is able to read NULL values.

<b>BTL - 7.3.2.13.X5 - Present Value Tracking Test</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Set the Group_Members property to make the Present_Value read a NULL value.
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	

### 3.X36.11 Is able to read Double values

The IUT supports a Global Group object that is able to read Double values.

<b>BTL - 7.3.2.13.X5 - Present Value Tracking Test</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Set the Group_Members property to make the Present_Value read a Double value.
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	

### 3.X36.12 Is able to read Character String values

The IUT supports a Global Group object that is able to read Character String values.

<b>BTL - 7.3.2.13.X5 - Present Value Tracking Test</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Set the Group_Members property to make the Present_Value read a Character String value.
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	

### 3.X36.13 Is able to read Octet String values

The IUT supports a Global Group object that is able to read Octet String values.

<b>BTL - 7.3.2.13.X5 - Present Value Tracking Test</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Set the Group_Members property to make the Present_Value read an Octet String value.
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	

### 3.X36.14 Is able to read Date values

The IUT supports a Global Group object that is able to read Date values.

<b>BTL - 7.3.2.13.X5 - Present Value Tracking Test</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .

<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Set the Group_Members property to make the Present_Value read a Date value.
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	

### 3.X36.15 Is able to read Time values

The IUT supports a Global Group object that is able to read Time values.

<b>BTL - 7.3.2.13.X5 - Present Value Tracking Test</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Set the Group_Members property to make the Present_Value read a Time value.
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	

### 3.X36.16 Is able to read BACnetObjectIdentifier values

The IUT supports a Global Group object that is able to read BACnetObjectIdentifier values.

<b>BTL - 7.3.2.13.X5 - Present Value Tracking Test</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Set the Group_Members property to make the Present_Value read a BACnetObjectIdentifier value.
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	

### 3.X36.17 Supports COV to retrieve remote property values

The IUT can be made to use COV to retrieve values for the Global Group object.

Requires that COV\_Resubscription\_Interval be present.

<b>BTL - 7.3.1.7.X1 - COV Resubscription Interval Test</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	

### 3.X36.18 Supports COVU\_Period and COVU\_Recipients properties

The IUT contains, or can be made to contain, a Global Group object that can generate UnconfirmedCOVNotifications using the COVU\_Period and COVU\_Recipients properties.

<b>Verify Checklist</b>	
<b>Test Method</b>	Manual



	<b>Configuration</b>	
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Verify that the IUT claims support for DS-COVU-B in the Checklist.
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	
<b>Verify Checklist</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Verify that the IUT claims support for DM-DDB-A in the Checklist.
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	
<b>BTL - 9.22.1.X2 - Writing to Properties Based on Data Type</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	The property that is written for this instance of this test shall be the COVU Recipients property.
	<b>Testing Hints</b>	Verify that any legal value of BACnetRecipient (as a list element) can be written.
	<b>Notes &amp; Results</b>	
<b>BTL - 9.22.1.X2 - Writing to Properties Based on Data Type</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	The property that is written for this instance of this test shall be the COVU Period property.
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	
<b>BTL - 7.3.2.13.X6 - Global Group COVU Initiation Test</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	Verify that any legal value of BACnetDestination (as a list element) can be written.
	<b>Notes &amp; Results</b>	
<b>BTL - 7.3.2.13.X6 - COVU Period and COVU Recipients Zero Test</b>		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	
<b>BTL - 8.3.X1 - COVU Recipients Notifications</b>		
	<b>Test Method</b>	Manual

<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	

### 3.X36.19 Supports Reliability Evaluation

The IUT contains, or can be made to contain, a Global Group object that can generate ConfirmedEventNotifications and UnconfirmedEventNotifications with an Event\_Type of CHANGE\_OF\_RELIABILITY.

<b>BTL - 8.4.X7.5 - CHANGE OF RELIABILITY with the FAULT STATUS FLAGS Algorithm</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	This test must be executed.
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	
<b>BTL - 8.4.X9 - CHANGE OF RELIABILITY with COMMUNICATION FAILURE Algorithm</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	This test must be executed.
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	

[In BTL Test Plan, add CHANGE\_OF\_STATUS\_FLAGS tests in section 5.2.X]

### 5.2.X1 Implements the CHANGE\_OF\_STATUS\_FLAGS Algorithm

The IUT contains, or can be made to contain, an object that can generate ConfirmedEventNotifications and UnconfirmedEventNotifications with an Event\_Type of CHANGE\_OF\_STATUS\_FLAGS.

<b>BTL - 8.4.X8 - CHANGE OF STATUS_FLAGS Test (ConfirmedEventNotification)</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	This test must be repeated once for each object type that is capable of generating event notifications with an Event_Type of CHANGE_OF_STATUS_FLAGS.
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	
<b>BTL - 8.5.X8 - CHANGE OF STATUS_FLAGS Test (UnconfirmedEventNotification)</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	This test must be repeated once for each object type that is capable of generating event notifications with an Event_Type of CHANGE_OF_STATUS_FLAGS.
<b>Testing Hints</b>	

<b>Notes &amp; Results</b>	
----------------------------	--

[In BTL Test Plan, add CHANGE\_OF\_STATUS\_FLAGS tests in section 5.3.X]

### 5.3.X1 Implements the CHANGE\_OF\_STATUS\_FLAGS Algorithm

The IUT contains, or can be made to contain, an Event Enrollment object that can generate ConfirmedEventNotifications and UnconfirmedEventNotifications with an Event\_Type of CHANGE\_OF\_STATUS\_FLAGS.

<b>BTL - 8.4.X8 - CHANGE OF STATUS_FLAGS Test (ConfirmedEventNotification)</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	This test must be repeated once for each object type that is capable of generating event notifications with an Event_Type of CHANGE OF STATUS_FLAGS.
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	
<b>BTL - 8.5.X8 - CHANGE OF STATUS_FLAGS Test (UnconfirmedEventNotification)</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	This test must be repeated once for each object type that is capable of generating event notifications with an Event_Type of CHANGE OF STATUS_FLAGS.
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	

[In BTL Test Plan, Change linked Test to reference BTL test]

### 7.4.15 Supports COV Logging of Remote Property Values

The IUT can be made to use COV logging.

Requires that COV\_Resubscription\_Interval be present.

<b><del>135.1-2013 - 7.3.2.24.5</del>BTL - 7.3.1.7.X1 - COV Resubscription Interval Test</b>	
<b>Test Method</b>	Manual
<b>Configuration</b>	As per <del>ASHRAE 135.1-2013</del> <i>BTL Specified Tests</i> .
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	
<b>Notes &amp; Results</b>	

[In BTL Specified Tests, add Global Group object tests in section 7.3.2.13]

### 7.3.2.13.X1 Global Group Present\_Value, Out\_Of\_Service and Status\_Flags Test

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

Purpose: This test verifies the interrelationship between the Present\_Value, Out\_Of\_Service and Status\_Flags properties of a Global Group object.

Test Concept: Verify the Present\_Value stops updating when Out\_Of\_Service is TRUE.

Configuration Requirements: The IUT shall be configured with a Global Group object with the Group\_Members property containing a member M1 at index N1 that has a value that can be changed. W1 is the maximum time it takes for the Global Group to receive an update from M1.

Test Steps:

1. MAKE (Out\_Of\_Service = TRUE)
2. VERIFY Out\_Of\_Service = TRUE
3. VERIFY Status\_Flags = {?, ?, FALSE, TRUE}
4. X1 = READ Present\_Value, ARRAY\_INDEX = N1
5. MAKE (M1 value change)
6. WAIT (W1)
7. X2 = READ Present\_Value, ARRAY\_INDEX = N1
8. VERIFY X1 = X2

#### **7.3.2.13.X2 Reliability MEMBER\_FAULT Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

Purpose: This test case verifies the FAULT flag of the Member\_Status\_Flags is TRUE and the Reliability property is equal to MEMBER\_FAULT when a member of the Group\_Members property goes into FAULT.

Test Concept: Force a member of the Group\_Members property to enter a Fault condition and verify the Member\_Status\_Flags FAULT flag equals TRUE and Reliability equals MEMBER\_FAULT.

Configuration Requirements: The IUT shall be configured with a Global Group object with the Group\_Members property containing a member M1 at index N1 that has a value that can be made to indicate a fault condition (see Notes To Tester). The Out\_Of\_Service property of the Global Group object must remain FALSE throughout the test. W1 is the maximum time it takes for the Global Group to receive an update from M1.

Test Steps:

1. MAKE (M1 Status\_Flags = {?, TRUE, ?, ?})
2. WAIT (W1)
3. VERIFY Member\_Status\_Flags = {?, TRUE, ?, ?}
4. VERIFY Reliability = MEMBER\_FAULT

Notes to Tester: Member\_Status\_Flags FAULT flag will the TRUE and the Reliability property will change to MEMBER\_FAULT when a member of the Group\_Members property goes into fault.

#### **7.3.2.13.X3 Reliability COMMUNICATION\_FAILURE Test**

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

Purpose: This test case verifies that the Member\_Status\_Flags FAULT flag will remain FALSE while the Reliability property is COMMUNICATION\_FAILURE.

Test Concept: Force a member of the Group\_Members property to stop communicating and verify the Reliability property equals COMMUNICATION\_FAILURE and the Member\_Status\_Flags FAULT flag remains FALSE.

Configuration Requirements: The IUT shall be configured with a Global Group object with the Group\_Members containing a member M1 at index N1 that can be made to discontinue communications and also respond with an error such as OBJECT/UNKNOWN\_OBJECT. (See Notes To Tester). The Out\_Of\_Service property of the Global Group object must remain FALSE throughout the test. W1 is the maximum time it takes for the Global Group to receive an update from M1.

Test Steps:

1. MAKE (M1 fail (communications or error))
2. WAIT (W1)
3. VERIFY Reliability = COMMUNICATION\_FAILURE
4. VERIFY Member\_Status\_Flags = {?, FALSE, ?, ?}

Notes to Tester: Reliability will change to COMMUNICATION\_FAILURE when a member is no longer able to communicate its Status\_Flags property. This can occur when the device goes offline or the object is deleted within the device.

#### **7.3.2.13.X4 Present\_Value Tracking and Reliability Test**

Dependencies: ReadProperty Service Execution Tests, 9.18

Purpose: This test verifies that the Global Group object continues to update its Present\_Value independent of the state of the Reliability property.

Test Concept: While the Reliability property is not NO\_FAULT\_DETECTED verify the Present\_Value continues to update.

Configuration Requirements: The IUT shall be configured with a Global Group object with its Reliability not equal to NO\_FAULT\_DETECTED and a Group\_Members member M1 at index N1 that can be changed. W1 is the maximum time it takes for the Global Group to receive an update from M1.

1. VERIFY Reliability <> NO\_FAULT\_DETECTED
2. MAKE (M1 = X1)
3. WAIT (W1)
4. X2 = READ Present\_Value, ARRAY\_INDEX = N1
5. VERIFY X1 = X2

Note to Tester: Reliability will change to COMMUNICATION\_FAILURE when a member is no longer able to communicate its Status\_Flags property. This can occur when the device goes offline or the object is deleted within the device. Also, the Reliability property will change to MEMBER\_FAULT when a member of the Group\_Members property goes into fault.

#### **7.3.2.13.X5 Present\_Value Tracking Test**

Dependencies: ReadProperty Service Execution Tests, 9.18

Purpose: This test verifies that the Global Group object tracks the value of the monitored properties value and data type.

Test Concept: Make a member of the Group\_Members property change value and verify the Present\_Value updates to match that value.

Configuration Requirements: The IUT shall be configured with a Global Group object with the Group\_Members containing a member M1 at index N1 of the specified data type that can be changed. W1 is the maximum time it takes for the Global Group to receive an update from M1.

1. MAKE (M1 = X1)
2. WAIT (W1)
3. X2 = READ Present\_Value, ARRAY\_INDEX = N1
4. VERIFY X1 = X2

### 7.3.2.13.X6 COVU\_Period and COVU\_Recipient Zero Test

Reason for Change: This test is not specified in any SSPC proposal.

Purpose: To verify that object O1 does not initiate UnconfirmedCOVNotification service requests when COVU\_Period is zero or COVU\_Recipient contains an empty list.

Test Concept: Configure O1 to produce unsubscribed UnconfirmedCOVNotifications, set COVU\_Period to zero and attempt to produce unsubscribed UnconfirmedCOVNotifications. Repeat with COVU\_Recipients containing an empty list.

Test Steps:

1. MAKE (O1 issue an unsubscribed UnconfirmedCOVNotification)
2. BEFORE Notification Fail Time
  - RECEIVE UnconfirmedCOVNotification-Request,
  - DESTINATION = (any valid address),
  - 'Subscriber Process Identifier' = 0,
  - 'Initiating Device Identifier' = IUT,
  - 'Monitored Object Identifier' = O1,
  - 'Time Remaining' = 0,
  - 'List of Values' = (any valid set of values)
3. MAKE (COVU\_Period = 0)
4. MAKE (O1 issue an unsubscribed UnconfirmedCOVNotification)
5. WAIT Notification Fail Time times 2
6. CHECK (Verify that O1 has not transmitted an UnconfirmedCOVNotification-Request.)
7. MAKE (COVU\_Period <> 0)
8. MAKE (O1 issue an unsubscribed UnconfirmedCOVNotification)
9. BEFORE Notification Fail Time
  - RECEIVE UnconfirmedCOVNotification-Request,
  - DESTINATION = (any valid address),
  - 'Subscriber Process Identifier' = 0,
  - 'Initiating Device Identifier' = IUT,
  - 'Monitored Object Identifier' = O1,
  - 'Time Remaining' = 0,
  - 'List of Values' = (any valid set of values)
10. MAKE (COVU\_Recipient an empty list)
11. MAKE (O1 issue an unsubscribed UnconfirmedCOVNotification)
12. WAIT Notification Fail Time times 2
13. CHECK (Verify that O1 has not transmitted an UnconfirmedCOVNotification-Request.)

[In BTL Specified Tests, add ConfirmedEventNotification tests in section 8.4]

### 8.4.X8 CHANGE\_OF\_STATUS\_FLAGS Test (ConfirmedEventNotification)

Reason for Change: New algorithm for Protocol\_Revision 13.

Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: <update these as appropriate>

Purpose: To verify the correct operation of the CHANGE\_OF\_STATUS\_FLAGS event algorithm. This test applies to objects that support an Event\_Type of CHANGE\_OF\_STATUS\_FLAGS.

Test Concept: The object O1 begins the test in a NORMAL state. The pMonitoredValue is changed such that a logical AND of pMonitoredValue and pSelectedFlags results in at least one bits set. After pTimeDelay expires the object shall enter the OFFNORMAL state and transmit an event notification message. The pMonitoredValue is then changed such that a logical AND of pMonitoredValue and pSelectedFlags results in no bits set. After pTimeDelayNormal expires the object shall enter the NORMAL state and transmit an event notification message.

Configuration Requirements: The O1 shall be configured such that the Event\_Enable property has a value of TRUE for the TO-OFFNORMAL and TO-NORMAL transitions. The 'Issue Confirmed Notifications' parameter in the Recipient\_List of the configured Notification Class Issue\_Confirmed\_Notifications property shall have a value of TRUE. The Recipient\_List of the configured Notification Class shall contain recipients. The event-generating object shall be in a NORMAL state at the start of the test.

Test Steps:

1. VERIFY Event\_State = NORMAL
2. MAKE (pMonitoredValue AND pSelectedFlags <> {FALSE, FALSE, FALSE, FALSE})
3. WAIT (pTimeDelay)
4. BEFORE **Notification Fail Time**
  - RECEIVE ConfirmedEventNotification-Request,
    - 'Process Identifier' = (any valid process ID),
    - 'Initiating Device Identifier' = IUT,
    - 'Event Object Identifier' = O1,
    - 'Time Stamp' = (the current local datetime or time or sequence number),
    - 'Notification Class' = (the notification class configured for O1),
    - 'Priority' = (the value configured for the transition),
    - 'Event Type' = CHANGE\_OF\_STATUS\_FLAGS,
    - 'Notify Type' = EVENT | ALARM,
    - 'Message Text' = (any valid message text),
    - 'AckRequired' = TRUE | FALSE,
    - 'From State' = NORMAL,
    - 'To State' = OFFNORMAL,
    - 'Event Values' = pPresentValue, pMonitoredValue
5. TRANSMIT BACnet-SimpleACK-PDU
6. VERIFY Status\_Flags = {TRUE, FALSE,?,?}
7. VERIFY Event\_State = OFFNORMAL
8. MAKE (pMonitoredValue AND pSelectedFlags = {FALSE, FALSE, FALSE, FALSE})
9. WAIT (pTimeDelayNormal)
10. BEFORE **Notification Fail Time**
  - RECEIVE ConfirmedEventNotification-Request,
    - 'Process Identifier' = (any valid process ID),
    - 'Initiating Device Identifier' = IUT,
    - 'Event Object Identifier' = O1
    - 'Time Stamp' = (the current local datetime or time or sequence number),
    - 'Notification Class' = (the notification class configured for O1),
    - 'Priority' = (the value configured for the transition),
    - 'Event Type' = CHANGE\_OF\_STATUS\_FLAGS,
    - 'Notify Type' = EVENT | ALARM,
    - 'Message Text' = (any valid message text),
    - 'AckRequired' = TRUE | FALSE,
    - 'From State' = OFFNORMAL,
    - 'To State' = NORMAL,
    - 'Event Values' = pPresentValue, pMonitoredValue
11. TRANSMIT BACnet-SimpleACK-PDU
12. VERIFY Status\_Flags = {FALSE, FALSE, ?, ?}
13. VERIFY Event\_State = NORMAL

#### 8.4.X9 CHANGE\_OF\_RELIABILITY with COMMUNICATION\_FAILURE Algorithm

Reason for Change: New fault.

Purpose: To verify the correct operation of the COMMUNICATION\_FAILURE fault.

Test Concept: The object under test (O1) begins the test in a NORMAL state with pCurrentReliability equal to NO\_FAULT\_DETECTED. Changes are made that result in pCurrentReliability equal COMMUNICATION\_FAILURE and

an event notification message is transmitted. Changes are made that return pCurrentReliability to NO\_FAULT\_DETECTED and another event notification message is transmitted.

Test Configuration: O1 is configured to detect faults and to report those using unconfirmed event notifications. O1 is initially configured to have no fault conditions present, and Event\_State is NORMAL.

Test Steps:

1. VERIFY pCurrentReliability = NO\_FAULT\_DETECTED
2. MAKE (pCurrentReliability = COMMUNICATION\_FAILURE)
4. BEFORE **Notification Fail Time**
  - RECEIVE UnconfirmedEventNotification-Request,
    - 'Process Identifier' = (any valid process ID),
    - 'Initiating Device Identifier' = IUT,
    - 'Event Object Identifier' = O1,
    - 'Time Stamp' = (the current local datetime or time or sequence number),
    - 'Notification Class' = (the notification class configured for O1),
    - 'Priority' = (the value configured for the transition),
    - 'Event Type' = CHANGE\_OF\_RELIABILITY,
    - 'Notify Type' = EVENT | ALARM,
    - 'Message Text' = (any valid message text),
    - 'AckRequired' = TRUE | FALSE,
    - 'From State' = NORMAL,
    - 'To State' = FAULT,
    - 'Event Values' = { COMMUNICATION\_FAILURE,  
(?, TRUE, ?, ?),  
(any valid values)
5. VERIFY pCurrentReliability = COMMUNICATION\_FAILURE
6. VERIFY Status\_Flags = (?, TRUE, ?,?)
7. VERIFY Event\_State = FAULT
2. MAKE (pCurrentReliability = NO\_FAULT\_DETECTED)
9. BEFORE **Notification Fail Time**
  - RECEIVE UnconfirmedEventNotification-Request,
    - 'Process Identifier' = (any valid process ID),
    - 'Initiating Device Identifier' = IUT,
    - 'Event Object Identifier' = O1,
    - 'Time Stamp' = (the current local datetime or time or sequence number),
    - 'Notification Class' = (the notification class configured for O1),
    - 'Priority' = (the value configured for the transition),
    - 'Event Type' = CHANGE\_OF\_RELIABILITY,
    - 'Notify Type' = EVENT | ALARM,
    - 'Message Text' = (any valid message text),
    - 'AckRequired' = TRUE | FALSE,
    - 'From State' = FAULT,
    - 'To State' = NORMAL,
    - 'Event Values' = { COMMUNICATION\_FAILURE,  
(?, FALSE, ?, ?),  
(any valid values)
10. VERIFY pCurrentReliability = NO\_FAULT\_DETECTED
11. VERIFY Status\_Flags = (?, FALSE, ?, ?)
12. VERIFY Event\_State = NORMAL

[In BTL Specified Tests, add UnconfirmedEventNotification tests in section 8.5]

### 8.5.X8 CHANGE\_OF\_STATUS\_FLAGS Test (UnconfirmedEventNotification)

Reason for Change: New algorithm for Protocol\_Revision 13.



Dependencies: ReadProperty Service Execution Tests, 9.18; WriteProperty Service Execution Tests, 9.22.

BACnet Reference Clauses: <update these as appropriate>

Purpose: To verify the correct operation of the CHANGE\_OF\_STATUS\_FLAGS event algorithm. This test applies to objects that support an Event\_Type of CHANGE\_OF\_STATUS\_FLAGS.

Test Concept: The object O1 begins the test in a NORMAL state. The pMonitoredValue is changed such that a logical AND of pMonitoredValue and pSelectedFlags results in at least one bits set. After pTimeDelay expires the object shall enter the OFFNORMAL state and transmit an event notification message. The pMonitoredValue is then changed such that a logical AND of pMonitoredValue and pSelectedFlags results in no bits set. After pTimeDelayNormal expires the object shall enter the NORMAL state and transmit an event notification message.

Configuration Requirements: The O1 shall be configured such that the Event\_Enable property has a value of TRUE for the TO-OFFNORMAL and TO-NORMAL transitions. The Issue\_Confirmed\_Notifications property shall have a value of FALSE. The event-generating object shall be in a NORMAL state at the start of the test.

Test Steps: The test steps for this test case are identical to the test steps in 8.4.X1 except that the ConfirmedEventNotification requests are UnconfirmedEventNotification requests and the TD does not acknowledge receiving the notifications.

Notes to Tester: The passing results for this test case are identical to the ones in 8.4.X1 except that the event notifications shall be conveyed using an UnconfirmedEventNotification service request. The MAC address used for these messages shall be either a broadcast that reaches the local network of the TD or the MAC address of the TD.

[In BTL Specified Tests, move and renumber 7.3.2.24.5 (specific to Trend Logs only) to 7.3.1.7.X1

### **7.3.2.25.51.7.X1 COV\_Resubscription Interval Test**

Dependencies: Confirmed Notifications Subscription, 8.10.1.

BACnet Reference Clause: 12.25.10 and 12.50.15.

Purpose: To verify that a ~~Trend Log~~ object O1 acquiring data via COV notification reissues its subscription at the interval set by COV\_Resubscription\_Interval.

Test Concept: ~~The Trend Log~~ O1 is configured to acquire data from the TD by COV notification. The TD verifies the resubscription interval.

Configuration Requirements: ~~Start\_Time, if present, shall be configured with a date and time preceding the beginning of the test. Stop\_Time, if present shall be configured with the latest possible date and time, in order that it occur after the end of the test. Stop\_When\_Full, if configurable, shall be set to FALSE. Enable shall be set to TRUE.~~ O1 is configured to acquire data from TD by COV notification. Non-zero values shall be chosen for COV\_Resubscription\_Interval in accordance with the range and resolution specified by the manufacturer for this property.

Test Steps:

1. IF (the IUT uses SubscribeCOV) THEN

RECEIVE SubscribeCOV-Request,  
'Subscriber Process Identifier' = (SPII, any value),  
'Monitored Object Identifier' = (MOII, the object to be monitored),  
'Issue Confirmed Notifications' = (ICNI = TRUE | FALSE),  
'Lifetime' = (LI, any value >= COV\_Resubscription\_Interval)

ELSE

RECEIVE SubscribeCOVProperty-Request,  
'Subscriber Process Identifier' = (SPII, any value),  
'Monitored Object Identifier' = (MOII, the object to be monitored),  
'Issue Confirmed Notifications' = (ICNI = TRUE | FALSE),  
'Lifetime' = (LI, any value >= COV\_Resubscription\_Interval),  
'Monitored Property Identifier' = (MPII, the property to be monitored),

- 'COV Increment' = (CII, Client\_COV\_Increment -- optional)
2. TRANSMIT BACnet-SimpleACK-PDU
  3. TRANSMIT ConfirmedCOVNotification-Request,
    - 'Subscriber Process Identifier' = SPII(~~corresponding value in step 1~~),
    - 'Initiating Device Identifier' = (~~Device object identifier of the TD~~),
    - 'Monitored Object Identifier' = MOII(~~corresponding value in step 1~~),
    - 'Issue Confirmed Notifications' = ICNI(~~corresponding value in step 1~~),
    - 'Time Remaining' = (any value <= ~~the Lifetime from step 1~~L1),
    - 'List of Values' = (appropriate BACnetPropertyValue(s))
  4. RECEIVE BACnet-SimpleACK-PDU
  5. BEFORE (the lesser of COV\_Resubscription\_Interval + **Re-subscription Interval Tolerance** and ~~L/LifeTime from step 4~~)
    - IF (the IUT uses SubscribeCOV)
      - RECEIVE SubscribeCOV-Request,
        - 'Subscriber Process Identifier' = SPII(~~corresponding value in step 1~~),
        - 'Monitored Object Identifier' = MOII(~~corresponding value in step 1~~),
        - 'Issue Confirmed Notifications' = ICNI(~~corresponding value in step 1~~),
        - 'Lifetime' = (L2, any value >= COV\_Resubscription\_Interval)
      - ELSE
        - RECEIVE SubscribeCOVProperty-Request,
          - 'Subscriber Process Identifier' = SPII(~~corresponding value in step 1~~),
          - 'Monitored Object Identifier' = MOII(~~corresponding value in step 1~~),
          - 'Issue Confirmed Notifications' = ICNI(~~corresponding value in step 1~~),
          - 'Lifetime' = (L2, any value >= COV\_Resubscription\_Interval)
          - 'Monitored Property Identifier' = MPPI(~~corresponding value in step 1~~),
          - 'COV Increment' = CII(~~corresponding value in step 1~~)
      - 6. TRANSMIT BACnet-SimpleACK-PDU
      - 7. TRANSMIT ConfirmedCOVNotification-Request,
        - 'Subscriber Process Identifier' = SPII(~~corresponding value in step 1~~),
        - 'Initiating Device Identifier' = (~~Device object identifier of the TD~~),
        - 'Monitored Object Identifier' = MOII(~~corresponding value in step 1~~),
        - 'Issue Confirmed Notifications' = ICNI(~~corresponding value in step 1~~),
        - 'Time Remaining' = (any value <= ~~the Lifetime from step 1~~L2),
        - 'List of Values' = (appropriate BACnetPropertyValue(s))
      - 8. RECEIVE BACnet-SimpleACK-PDU
      - 9. WAIT (COV\_Resubscription\_Interval - **Re-subscription Interval Tolerance**)
      - 10. BEFORE (2 \* **Re-subscription Interval Tolerance**)
        - IF (the IUT uses SubscribeCOV)
          - RECEIVE SubscribeCOV-Request,
            - 'Subscriber Process Identifier' = SPII(~~corresponding value in step 1~~),
            - 'Monitored Object Identifier' = MOII(~~corresponding value in step 1~~),
            - 'Issue Confirmed Notifications' = ICNI(TRUE),
            - 'Lifetime' = L1(~~corresponding value in step 1~~)
          - ELSE
            - RECEIVE SubscribeCOVProperty-Request,
              - 'Subscriber Process Identifier' = SPII(~~corresponding value in step 1~~),
              - 'Monitored Object Identifier' = MOII(~~corresponding value in step 1~~),
              - 'Issue Confirmed Notifications' = ICNI(TRUE),
              - 'Lifetime' = L1(~~corresponding value in step 1~~)
              - 'Monitored Property Identifier' = MPPI(~~corresponding value in step 1~~),
              - 'COV Increment' = CII(~~corresponding value in step 1~~)
        - 11. TRANSMIT BACnet-SimpleACK-PDU

Passing Result: Where the Lifetime parameter of a SubscribeCOV request is less than COV\_Resubscription\_Interval + Re-subscription Interval Tolerance, the IUT shall send the subsequent SubscribeCOV request within Lifetime seconds even though this is a smaller time window than defined by the test. If the IUT does not meet this stricter time window, then the IUT shall fail the test.

[In BTL Specified Tests, add 8.3.X1 to UnconfirmedCOVNotifications]

**8.3.X1 COVU\_Recipients Notifications**

Purpose: To verify that the IUT initiates UnconfirmedCOVNotification service requests to each entry in its COVU\_Recipients property based on COVU\_Period.

Test Concept: The IUT contains a Global Group object, O1, that is configured to periodically send UnconfirmedCOVNotification using COVU\_Period and COVU\_Recipients. The TD checks for these notifications.

Configuration Requirements: COVU\_Recipients property shall be non-empty and contain at least one device and one address based recipient. The COVU\_Period shall be non-zero.

Test Steps:

1. REPEAT X = (each entry in the COVU\_Recipients) DO {
  - BEFORE COVU\_Period + **Notification Fail Time**
  - RECEIVE UnconfirmedCOVNotification-Request,
    - DESTINATION = X,
    - 'Subscriber Process Identifier' = 0,
    - 'Initiating Device Identifier' = IUT,
    - 'Monitored Object Identifier' = O1,
    - 'Time Remaining' = 0,
    - 'List of Values' = (Member\_Status\_Flags,  
Elements of Present\_Value)
2. READ T1 = Local\_Time
3. REPEAT X = (each entry in the COVU\_Recipients) DO {
  - BEFORE COVU\_Period + **Notification Fail Time**
  - RECEIVE UnconfirmedCOVNotification-Request,
    - DESTINATION = X,
    - 'Subscriber Process Identifier' = 0,
    - 'Initiating Device Identifier' = IUT,
    - 'Monitored Object Identifier' = O1,
    - 'Time Remaining' = 0,
    - 'List of Values' = (Member\_Status\_Flags,  
Elements of Present\_Value)
4. READ T2 = Local\_Time
5. CHECK (T2 - T1 ~≠ COVU\_Period)

Note to tester: The test shall pass regardless of the order in which the IUT generates the UnconfirmedCOVNotification-Requests in each step.