



**BACnet[®] TESTING LABORATORIES
ADDENDA**

**Addendum o to
BTL Test Package 15**

**Revision 7
Revised 8/7/2019**

Approved by the BTL Working Group on May 23, 2019;
Approved by the BTL Working Group Voting Members on August 6, 2019;
Published on August 15, 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-15.2o-1: Fix Event Log Tests (BTLWG-18. BTLWG-325)2

BTL-15.2o-2: Changes to AE-AVM-A and AE-VM-A Test Plan (BTLWG-100).....7

BTL-15.2o-3: Clarify Calendar Entry WeekNDay Day Of Week Test (BTLWG-200).....9

BTL-15.2o-4: Clarify the Limit_Enable Tests (BTLWG-322).....10

BTL-15.2o-5: Clarify the Expected Behavior for COV Notifications (BTLWG-402)13

BTL-15.2o-6: Add Array Resizing Test for WPM Service (BTLWG-444).....14

BTL-15.2o-7: Remove Section 9.4.7 Is Able to Register as Foreign Device in BBMD (BTLWG-550).....16

BTL-15.2o-8: Remove Hop-Count Protection from Virtual Tests (BTLWG-271)18

BTL-15.2o-9: Modify Pulse Converter Test to Correctly Indicate How to Change Status_Flags (BTLWG-461).....19

BTL-15.2o-10: Move COV-A Negative Tests (BTLWG-54)21

BTL-15.2o-11: Clarify Internally Written Datatypes Test (BTLWG-274).....23

BTL-15.2o-12: Update Device Restart Test for Sequence Numbers (BTLWG-459)25

BTL-15.2o-13: Update Schedule Test Directives for Single Exception Schedule Devices (BTLWG-598)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-15.20-1: Fix Event Log Tests (BTLWG-18, BTLWG-325)

Overview:

Event Log object tests assembled into 135.1-2011 made some editing errors and omissions, pointed out in CR-0400. This proposal also fixes Tests 7.3.2.25.1, 7.3.2.26, and 7.3.2.28 which had incorrect Result Flags.

Changes:

[In BTL Specified Tests, add the following tests as modified from version of the tests from 135.1-2013]

7.3.2.25 Event Log Tests

The tests in this section verify that Event Log objects correctly record event notifications.

Some of the general logging object tests in Clause 7.3.2.24 are also applicable to the Event Log object type.

7.3.2.25.1 Internal Logging of Notifications

Purpose: To verify the IUT correctly collects and represents the Notifications which it initiates.

Test Concept: Make the IUT generate two event notification messages which the IUT logs. Use ReadRange to retrieve them from an Event Log and compare the two representations.

Configuration Requirements: The tester shall choose two events which are configured to be sent to the TD and to be placed into one of the IUT's Event Logs, LO1.

Test Steps:

1. WRITE Enable = TRUE
2. MAKE (IUT generate an EventNotification)
3. RECEIVE ConfirmedEventNotification-Request,
 - 'Process Identifier' = (any valid process identifier),
 - 'Initiating Device Identifier' = IUT,
 - 'Event Object Identifier' = (any valid object),
 - 'Time Stamp' = (T1, any valid timestamp),
 - 'Notification Class' = (any valid notification class),
 - 'Priority' = (any valid priority),
 - 'Event Type' = (any standard event type),
 - 'Message Text' = (any character string),
 - 'Notify Type' = ALARM | EVENT,
 - 'AckRequired' = TRUE | FALSE,
 - 'From State' = (state S1, any valid state for this event type),
 - 'To State' = (state S2, any valid state for this event type that can follow S1),
 - 'Event Values' = (any values appropriate to the event type)
4. TRANSMIT BACnet-SimpleACK-PDU
5. MAKE (IUT generate an EventNotification)
6. RECEIVE ConfirmedEventNotification-Request,
 - 'Process Identifier' = (any valid process identifier),
 - 'Initiating Device Identifier' = IUT,
 - 'Event Object Identifier' = (any valid object),
 - 'Time Stamp' = (T2, any valid timestamp),
 - 'Notification Class' = (any valid notification class),
 - 'Priority' = (any valid priority),
 - 'Event Type' = (any standard event type),
 - 'Message Text' = (any character string),
 - 'Notify Type' = ALARM | EVENT,
 - 'AckRequired' = TRUE | FALSE,
 - 'From State' = (state S3, any valid state for this event type),
 - 'To State' = (state S4, any valid state for this event type that can follow S3),
 - 'Event Values' = (any values appropriate to the event type)

7. TRANSMIT BACnet-SimpleACK-PDU
8. READ RC = LO1, Record_Count
9. TRANSMIT ReadRange-Request,
 - 'Object Identifier' = LO1,
 - 'Property Identifier' = Log_Buffer,
 - 'Reference Index' = RC,
 - 'Count' = -2
10. RECEIVE ReadRange-ACK,
 - 'Object Identifier' = LO1,
 - 'Property Identifier' = Log_Buffer,
 - 'Result Flags' = {FALSE, ?, FALSETRUE},
 - 'Item Count' = 2,
 - 'Item Data' = (logged data that matches the information received in steps 3 and 6, except that Process_Identifier may be any value and is not required to match)
11. CHECK (T2 > T1, and that the notifications were logged in order)

Notes to Tester: When the UnconfirmedEventNotification service is used instead of the ConfirmedEventNotification service, the TD shall skip the steps in which a SimpleACK-PDU is sent.

7.3.2.26 Remote Logging of Notifications

Purpose: To verify that the IUT correctly collects and represents the Notifications which it receives.

Test Concept: Make TD send multiple event notification messages. Use ReadRange to retrieve the events from an Event Log or perhaps from multiple Event Logs in the IUT, and compare the two representations.

Configuration Requirements: LO1 is an Event Log object in IUT which logs the event types which are sent. Stop_When_Full in LO1 shall be FALSE or absent.

Test Steps:

1. WRITE Enable = TRUE
2. TRANSMIT ConfirmedEventNotification-Request,
 - 'Process Identifier' = (any valid process identifier),
 - 'Initiating Device Identifier' = TD,
 - 'Event Object Identifier' = (any valid object identifier),
 - 'Time Stamp' = (T1, any valid timestamp),
 - 'Notification Class' = (any valid notification class),
 - 'Priority' = (any valid priority),
 - 'Event Type' = (any standard event type),
 - 'Message Text' = (any character string),
 - 'Notify Type' = ALARM | EVENT,
 - 'AckRequired' = TRUE | FALSE,
 - 'From State' = (state S1, any valid state for this event type),
 - 'To State' = (state S2, any valid state for this event type that can follow S1),
 - 'Event Values' = (any values appropriate to the event type)
3. RECEIVE BACnet-SimpleACK-PDU
4. TRANSMIT ConfirmedEventNotification-Request,
 - 'Process Identifier' = (any valid process identifier),
 - 'Initiating Device Identifier' = IUT,
 - 'Event Object Identifier' = (any valid object identifier),
 - 'Time Stamp' = (T2, any valid timestamp),
 - 'Notification Class' = (any valid notification class),
 - 'Priority' = (any valid priority),
 - 'Event Type' = (any standard event type),
 - 'Message Text' = (any character string),
 - 'Notify Type' = ALARM | EVENT,
 - 'AckRequired' = TRUE | FALSE,
 - 'From State' = (state S3, any valid state for this event type),
 - 'To State' = (state S4, any valid state for this event type that can follow S3),
 - 'Event Values' = (any values appropriate to the event type)

5. RECEIVE BACnet-SimpleACK-PDU
6. READ RC = LO1, Record_Count
7. TRANSMIT ReadRange-Request,
 - 'Object Identifier' = LO1,
 - 'Property Identifier' = Log_Buffer,
 - 'Reference Index' = RC,
 - 'Count' = -2
8. RECEIVE ReadRange-ACK,
 - 'Object Identifier' = LO1,
 - 'Property Identifier' = Log_Buffer,
 - 'Result Flags' = {FALSE, ?, FALSETRUE},
 - 'Item Count' = 2,
 - 'Item Data' = (logged data that matches the information received in steps 2 and 4, except that Process_Identifier can be any value and is not required to match)
9. CHECK (that the events were logged in the order in which they were received)

Notes to Tester: When the UnconfirmedEventNotification service is used instead of the ConfirmedEventNotification service, the test shall skip the steps in which a BACnet-SimpleACK-PDU is expected.

7.3.2.27 Internal Logging of ACK_NOTIFICATIONs

Purpose: To verify the IUT correctly collects and represents an ACK_NOTIFICATION which it initiates.

Test Concept: Make the IUT generate an ACK_NOTIFICATION message. Use ReadRange to retrieve that same event from an Event Log and compare the two representations. If the IUT does not support logging of the ACK_NOTIFICATIONs which it initiates, this test shall be skipped.

Configuration Requirements: O1 is an event initiating object in the IUT, which is configured to send event notifications to TD. LO1 is an Event Log object in IUT which logs ACK_NOTIFICATIONs.

Test Steps:

1. WRITE Enable = TRUE
- ~~2. READ RC = LO1, Record_Count~~
23. MAKE (the IUT generate a notification)
34. RECEIVE ConfirmedEventNotification-Request,
 - 'Process Identifier' = (PI1, any valid process identifier),
 - 'Initiating Device Identifier' = IUT,
 - 'Event Object Identifier' = O1,
 - 'Time Stamp' = (T1, any valid timestamp),
 - 'Notification Class' = (N1, any valid notification class),
 - 'Priority' = (P1, any valid priority),
 - 'Event Type' = (ET1, any standard event type),
 - 'Message Text' = (any character string),
 - 'Notify Type' = ALARM | EVENT,
 - 'AckRequired' = TRUE | FALSE,
 - 'From State' = (S1, any valid state for this event type),
 - 'To State' = (S2, any valid state for this event type),
 - 'Event Values' = (any values appropriate to the event type)
45. TRANSMIT BACnet-SimpleACK-PDU
56. TRANSMIT AcknowledgeAlarm-Request,
 - 'Acknowledging Process Identifier' = (any valid value),
 - 'Event Object Identifier' = O1,
 - 'Event State Acknowledged' = S2,
 - 'Time Stamp' = T1,
 - 'Time of Acknowledgment' = (the current time)
- ~~67. RECEIVE BACnet-SimpleACK-PDU~~
78. BEFORE **Notification Fail Time**
 RECEIVE ConfirmedEventNotification-Request,
 - 'Process Identifier' = PI1,

'Initiating Device Identifier' = IUT,
 'Event Object Identifier' = O1,
 'Time Stamp' = (T2, any valid timestamp > T1),
 'Notification Class' = N1,
 'Priority' = P1,
 'Event Type' = ET1,
 'Message Text' = (any character string),
 'Notify Type' = ACK_NOTIFICATION,
 'From State' = S1

89. TRANSMIT BACnet-SimpleACK-PDU

9. *READ RC = LO1, Record_Count*

10. TRANSMIT ReadRange-Request,

'Object Identifier' = LO1,
 'Property Identifier' = Log_Buffer,
 'Reference Index' = RC,
 'Count' = -1

11. RECEIVE ReadRange-ACK,

'Object Identifier' = LO1,
 'Property Identifier' = Log_Buffer,
 'Result Flags' = {FALSE, ?, FALSETRUE},
 'Item Count' = 1,
 'Item Data' = (logged data that matches the information received in step 74,
 except that Process_Identifier can be any value and is not
 required to match)

Notes to Tester: When the UnconfirmedEventNotification service is used instead of the ConfirmedEventNotification service, the TD shall skip the steps in which BACnet-SimpleACK-PDUs are sent in response to ConfirmedEventNotifications.

7.3.2.28 Remote Logging of ACK_NOTIFICATIONS

Purpose: To verify that the IUT correctly collects and represents ACK_NOTIFICATIONS which it receives.

Test Concept: Send an ACK_NOTIFICATION to the IUT. Use ReadRange to retrieve that same event from an Event Log, and compare the two representations.

Configuration Requirements: LO1 is an Event Log object in IUT which logs ACK_NOTIFICATIONS. Stop_When_Full in LO1 shall be FALSE or absent.

Test Steps:

1. WRITE Enable = TRUE
2. TRANSMIT ConfirmedEventNotification-Request,
 'Process Identifier' = (any valid process identifier),
 'Initiating Device Identifier' = IUT,
 'Event Object Identifier' = (any valid object identifier),
 'Time Stamp' = (T1, any valid timestamp),
 'Notification Class' = (any valid notification class),
 'Priority' = (any valid priority),
 'Event Type' = (any standard event type),
 'Message Text' = (any character string),
 'Notify Type' = ACK_NOTIFICATION,
 'From State' = (state S1, any valid state for this event type)
3. RECEIVE BACnet-SimpleACK-PDU
4. READ RC = LO1, Record_Count
5. TRANSMIT ReadRange-Request,
 'Object Identifier' = LO1,
 'Property Identifier' = Log_Buffer,
 'Reference Index' = RC,
 'Count' = -1
6. RECEIVE ReadRange-ACK,
 'Object Identifier' = LO1,

'Property Identifier' = Log Buffer,
 'Result Flags' = {FALSE, ?, FALSE, TRUE},
 'Item Count' = 1,
 'Item Data' = (logged data that matches the information received in step 2,
 except that Process_Identifier can be any value and is not required to match)

Notes to Tester: When the UnconfirmedEventNotification service is used instead of the ConfirmedEventNotification service, the test shall skip the step in which a BACnet-SimpleACK-PDU is expected.

[In BTL Test Plan section 5.14.1, change test references from 135.1-2013 to BTL Specified Tests]

5.14 Alarm and Event Management - Event Log - Internal - B

5.14.1 Base Requirements

.....	
135.1-2013 BTL - 7.3.2.25.1 - Internal Logging of Notifications	
Test Conditionality	Must be executed.
Test Directives	REPEAT for both Confirmed and UnconfirmedEventNotifications..
Testing Hints	REPEAT for events with optional message text present, and with message text not present if the device supports both.

[In BTL Test Plan section 5.14.4, change test references from 135.1-2013 to BTL Specified Tests]

5.14.4 Supports logging of ACK_NOTIFICATION

135.1-2013 BTL - 7.3.2.27 - Internal Logging of ACK_NOTIFICATIONS	
Test Conditionality	Must be executed.
Test Directives	
Testing Hints	

[In BTL Test Plan section 5.15.4, change test references from 135.1-2013 to BTL Specified Tests]

5.15 Alarm and Event Management - Event Log - External - B

5.15.4 Supports Logging all Notifications

135.1-2013 BTL - 7.3.2.26 - Remote Logging of Notifications	
Test Conditionality	Must be executed.
Test Directives	REPEAT for both Confirmed and UnconfirmedEventNotifications. REPEAT for events which contain the optional message text parameter and for those which don't. REPEAT for all BACnetEventTypes up to the Protocol_Revision claimed by the IUT. TRANSMIT EventNotifications of a size that the IUT is capable of logging.
Testing Hints	This does include CHANGE_OF_LIFE_SAFETY, BUFFER_READY, and complex event types.
135.1-2013 BTL - 7.3.2.28 - Remote Logging of ACK_NOTIFICATIONS	
Test Conditionality	Must be executed.
Test Directives	TRANSMIT EventNotifications of a size that the IUT is capable of logging.
Testing Hints	

BTL-15.2o-2: Changes to AE-AVM-A and AE-VM-A Test Plan (BTLWG-100)

Overview:

135-2012bc-3 updated the AE-VM-A and AE-AVM-A BIBBs to show new event and fault algorithm parameters instead of specific properties of specific objects. Additionally in the case of AE-AVM-A, Notification Forwarder and Alert Enrollment objects were added to the BIBB.

Changes:

[In BTL Test Plan, update Conditionality and Hints for section AE-VM-A as shown.]

5.17 Alarm and Event Management- View Modify - A

5.17.1 Base Requirements

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

135.1-2013 - 8.18.3 - Reading and Presenting Properties		
	Test Conditionality	Must be executed. <i>Note: If the IUT also claims support for AE-AVM-A, this test may be omitted.</i>
	Test Directives	
	Testing Hints	Repeat the test for each <i>standard property</i> , in each <i>standard object type</i> , which represent parameters to an event or fault algorithm excluding those for algorithms excluded by the BIBB definition. of the following event generating objects: Analog Input/Output/Value, Accumulator, Event Enrollment, Loop, and Pulse Converter. Repeat for each of the required properties listed in the table in the BIBB definition.
135.1-2013 - 8.22.4 - Accepting Input and Modifying Properties		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	Repeat the test for each <i>standard property</i> , in each <i>standard object type</i> , which represent parameters to an event or fault algorithm excluding those for algorithms excluded by the BIBB definition. of the following event generating objects: Analog Input/Output/Value, Accumulator, Event Enrollment, Loop, and Pulse Converter. Repeat for each of the required properties listed in the table in the BIBB definition. Repeat the test for a variety of values that cover the range of values required by the BIBB.

[In BTL Test Plan, Update section 5.19 AE-AVM-A with the following conditionality and hints changes.]

5.19 Alarm and Event Management - Advanced View Modify - A

5.19.1 Base Requirements

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

135.1-2013 - 8.18.3 - Reading and Presenting Properties		
	Test Conditionality	Must be executed. <i>Note: If the IUT also claims support for AE-AVM-A, this test may be omitted.</i>
	Test Directives	
	Testing Hints	Repeat the test for each <i>standard property</i> , in each <i>standard object type</i> , which represent parameters to an event or fault algorithm excluding those for algorithms excluded by the BIBB definition. of the following event generating objects: Analog Input/Output/Value, Accumulator, Event Enrollment, Loop, and Pulse Converter. Repeat for each <i>property listed in the tables in the BIBB</i> , in each <i>standard object type</i> , excluding those for algorithms excluded by the

		BIBB definition. of the required properties listed in the table in the BIBB definition.
135.1-2013 - 8.22.4 - Accepting Input and Modifying Properties		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	Repeat the test for each <i>standard property</i> , in each <i>standard object type</i> , which represent parameters to an event or fault algorithm excluding those for algorithms excluded by the BIBB definition. of the following event generating objects: Analog Input/Output/Value, Accumulator, Event Enrollment, Loop, and Pulse Converter. Repeat for each <i>property listed in the tables in the BIBB</i> , in each <i>standard object type</i> , excluding those for algorithms excluded by the BIBB definition. of the required properties listed in the table in the BIBB definition. Repeat the test for a variety of values that cover the range of values required by the BIBB.

5.19.2 Supports DS-RP-A

The IUT shall support DS-RP-A in order to receive alarm parameters for presentation to the user.

Verify Checklist		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims support for DS-RP-A.
	Testing Hints	

5.19.3 Supports DS-WP-A

The IUT shall support DS-WP-A in order to update alarm parameters modified by the user.

Verify Checklist		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims support for DS-WP-A.
	Testing Hints	

5.19.4 Supports DM-OCD-A

The IUT shall support DS-OCD-A in order to allow the user to create Event Enrollment, ~~and~~ Notification Class, *and Notification Forwarder* objects.

Verify Checklist		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims support for DM-OCD-A and that it claims the ability to create and delete Event Enrollment, and Notification Class, <i>and Notification Forwarder</i> objects.
	Testing Hints	

BTL-15.2o-3: Clarify Calendar Entry WeekNDay Day Of Week Test (BTLWG-200)

Overview:

Added clarifying text to Table 7-16.1, Value column per BTL-CR-0406.

Changes:

[In BTL Test Plan, change all references of test 135-1-2013 - 7.3.2.23.20.3.7 to BTL - 7.3.2.23.20.3.7, there are 2 of them.]

[In BTL Specified Tests, add the following test with changes from 135.1 version shown]

7.3.2.23.10.3.7 Revision 4 Calendar Entry WeekNDay Day Of Week Test

Reason for Change: Added clarifying text to table 7-16.1.

Dependencies: ReadProperty Service Execution Tests, 9.18; TimeSynchronization Service Execution Tests, 9.30, UTCTimeSynchronization Service Execution Tests, 9.31.

BACnet Reference Clause: 12.24.8.

Purpose: To verify that a date matching a WeekNDay's DayOfWeek field in an Exception_Schedule enables the referencing Schedule object.

Test Concept: The IUT's local date and time are changed to values that are selected by the TD based on the criteria in Table 7-10. The value of the Present_Value property is monitored to verify that the scheduled write operations occur.

Configuration Requirements: The IUT shall be configured to contain a Schedule object with an Exception_Schedule containing a BACnetCalendarEntry with a WeekNDay entry specifying the day of the week. The criteria for the dates used in the test are given in Table 7-10. The local date and time shall be set such that the Present_Value property has a value other than V₁.

Table 7-16.1. Criteria for Calendar Entry WeekNDay Day of Week Test Dates and Values

Date	Criteria	Value
D ₁	1. Date occurs during Effective_Period, 2A. BACnetSpecialEvent incorporates calendarEntry: WeekNDay, 2B. calendarEntry: WeekNDay specifies only DayOfWeek, 2C. Date falls on the specified day of the week, and 2D. Higher eventPriority than any coincident BACnetSpecialEvents.	V ₁
D ₂	1. Date occurs during Effective_Period, 2A. BACnetSpecialEvent incorporates calendarEntry: WeekNDay, 2B. calendarEntry: WeekNDay specifies only DayOfWeek, and 2C. Date does not fall on the specified day of the week.	V ₂ <i>(different from any other values)</i>

Test Steps:

1. VERIFY Present_Value = (any value other than V₁)
2. (TRANSMIT TimeSynchronization-Request, 'Time' = D₁) |
 (TRANSMIT UTCTimeSynchronization-Request, 'Time' = D₁) |
 MAKE (the local date and time = D₁)
3. WAIT **Schedule Evaluation Fail Time**
4. VERIFY Present_Value = V₁
5. (TRANSMIT TimeSynchronization-Request, 'Time' = D₂) |
 (TRANSMIT UTCTimeSynchronization-Request, 'Time' = D₂) |
 MAKE (the local date and time = D₂)
6. WAIT **Schedule Evaluation Fail Time**
7. VERIFY Present_Value = (any value other than V₂)

BTL-15.2o-4: Clarify the Limit_Enable Tests (BTLWG-322)

Overview:

Clarify what the 'Event Type' parameter should be by making the Limit_Enable tests use 'Event Type' = the algorithm appropriate to the object type i.e. OUT_OF_RANGE, SIGNED_OUT_OF_RANGE, UNSIGNED_OUT_OF_RANGE, or DOUBLE_OUT_OF_RANGE.

Changes:

[In BTL Test Plan, modify test references for BTL - 7.3.1.13.X1 and BTL - 7.3.1.13.X2 to use the new test numbers of BTL - 7.3.1.13.1 and BTL - 7.3.1.13.2, there are 2 of each.]

[In BTL Specified Tests, revise existing tests 7.3.1.13.X1 and 7.3.1.13.X2 as shown. Also renumber to match the Addenda q version.]

7.3.1.13.X1 7.3.1.13.1 Limit_Enable Test, LowLimitEnable

Reason for Change: The 'Event Type' is checked to be an out-of-range algorithm appropriate to the object type.

Purpose: To verify that the LowLimitEnable flag in the Limit_Enable property correctly enables or disables reporting of out of range events. This test applies to objects with a Limit_Enable property.

Test Concept: The LowLimitEnable flag is set to true in the Limit_Enable property and the event-triggering property is manipulated to cause the low limit to be exceeded. This should generate an event notification and make Event_State = Low_Limit. After the event-triggering property is returned to a normal value, the LowLimitEnable flag is the set to false and the event-triggering property is again manipulated to exceed the low limit. No event notification should be observed and the Event_State must have a value of normal.

Configuration Requirements: Configure the object with pHighLimit, pLowLimit and pDeadband values such that pLowLimit + pDeadband < pHighLimit and both the pLowLimit and pHighLimit values are within the valid range of values for the event-triggering property. If the device cannot be configured with limit values that meet these conditions, then this test shall be skipped. The Event_Enable property shall be set to (TRUE, ?, TRUE) for this test. If the Event_Enable property cannot be configured such that the TO-NORMAL and the TO-OFFNORMAL transitions are TRUE, this test shall be skipped.

Test Steps:

1. MAKE pLimitEnable = (TRUE, ?)
2. VERIFY pCurrentState = NORMAL
3. MAKE (pMonitoredValue a value less than pLowLimit)
4. WAIT (pTimeDelay)
5. BEFORE Notification Fail Time
 - RECEIVE ConfirmedEventNotification-Request,
 - 'Process Identifier' = (any valid process ID),
 - 'Initiating Device Identifier' = IUT,
 - 'Event Object Identifier' = (the object configured for this test),
 - 'Time Stamp' = (the current local time),
 - 'Notification Class' = (the class corresponding to the object being tested),
 - 'Priority' = (the value configured to correspond to a TO-OFFNORMAL transition),
 - 'Event Type' = *(the algorithm appropriate to the object type i.e. OUT_OF_RANGE, SIGNED_OUT_OF_RANGE, UNSIGNED_OUT_OF_RANGE, or DOUBLE_OUT_OF_RANGE),*
 - 'Message Text' = (optional, any valid message text),
 - 'Notify Type' = ALARM | EVENT,
 - 'AckRequired' = TRUE | FALSE,
 - 'From State' = NORMAL,
 - 'To State' = LOW_LIMIT,
 - 'Event Values' = (values appropriate to the event type)
6. TRANSMIT SimpleAck-PDU
7. VERIFY pCurrentState = LOW_LIMIT
8. MAKE (pMonitoredValue a value that is between pLowLimit + pDeadband and pHighLimit)
9. WAIT (pTimeDelayNormal)

10. BEFORE Notification Fail Time

RECEIVE ConfirmedEventNotification-Request,
 'Process Identifier' = (any valid process ID),
 'Initiating Device Identifier' = IUT,
 'Event Object Identifier' = (the object configured for this test),
 'Time Stamp' = (the current local time),
 'Notification Class' = (the class corresponding to the object being tested),
 'Priority' = (the value configured to correspond to a TO-NORMAL transition),
 'Event Type' = *(the algorithm appropriate to the object type i.e. OUT_OF_RANGE, SIGNED_OUT_OF_RANGE, UNSIGNED_OUT_OF_RANGE, or DOUBLE_OUT_OF_RANGE),*
 'Message Text' = (optional, any valid message text),
 'Notify Type' = ALARM | EVENT,
 'AckRequired' = TRUE | FALSE,
 'From State' = LOW_LIMIT,
 'To State' = NORMAL,
 'Event Values' = (values appropriate to the event type)

11. TRANSMIT SimpleAck-PDU

12. MAKE pLimitEnable = (FALSE, ?)

13. VERIFY pCurrentState = NORMAL

14. MAKE (pMonitoredValue a value less than pLowLimit)

15. WAIT (pTimeDelay + Notification Fail Time)

16. CHECK (verify that no notification message was transmitted)

17. VERIFY pCurrentState = NORMAL

Notes to Tester: The UnconfirmedEventNotification service may be substituted for the ConfirmedEventNotification service in which case the TD shall skip all of the steps in which a SimpleACK-PDU is sent.

~~7.3.1.13.X~~7.3.1.13.2 Limit_Enable Test, HighLimitEnable

Reason for Change: The 'Event Type' is checked to be an out-of-range algorithm appropriate to the object type.

Purpose: To verify that the HighLimitEnable flag in the Limit_Enable property correctly enables or disables reporting of out of range events. This test applies to objects with a Limit_Enable property.

Test Concept: The HighLimitEnable flag is set to true in the Limit_Enable property and the event-triggering property is manipulated to cause the high limit to be exceeded. This should generate an event notification and make Event_State = High_Limit. After the event-triggering property is returned to a normal value, the HighLimitEnable flag is set to false and the event-triggering property is again manipulated to exceed the high limit. No event notification should be observed and the Event_State must have a value of normal.

Configuration Requirements: Configure the object with pHighLimit, pLowLimit and pDeadband values such that pHighLimit - pDeadband > pLowLimit and both the pLowLimit and pHighLimit values are within the valid range of values for the event triggering property. If the device cannot be configured with limit values that meet these conditions, then this test shall be skipped. The Event_Enable property shall be set to (TRUE, ?, TRUE) for this test. If the Event_Enable property cannot be configured such that the TO-NORMAL and the TO-OFFNORMAL transitions are TRUE, this test shall be skipped.

Test Steps:

1. MAKE pLimitEnable = (?, TRUE)

2. VERIFY pCurrentState = NORMAL

3. MAKE (pMonitoredValue a value that exceeds pHighLimit)

4. WAIT (pTimeDelay)

5. BEFORE Notification Fail Time

RECEIVE ConfirmedEventNotification-Request,
 'Process Identifier' = (any valid process ID),
 'Initiating Device Identifier' = IUT,
 'Event Object Identifier' = (the object configured for this test),
 'Time Stamp' = (the current local time),
 'Notification Class' = (the class corresponding to the object being tested),
 'Priority' = (the value configured to correspond to a TO-OFFNORMAL transition),
 'Event Type' = *(the algorithm appropriate to the object type i.e. OUT_OF_RANGE,*

- | | |
|------------------|---|
| | <i>SIGNED_OUT_OF_RANGE, UNSIGNED_OUT_OF_RANGE, or DOUBLE_OUT_OF_RANGE),</i> |
| 'Message Text' = | (optional, any valid message text), |
| 'Notify Type' = | ALARM EVENT, |
| 'AckRequired' = | TRUE FALSE, |
| 'From State' = | NORMAL, |
| 'To State' = | HIGH_LIMIT, |
| 'Event Values' = | (values appropriate to the event type) |
6. TRANSMIT SimpleAck-PDU
 7. VERIFY pCurrentState = HIGH_LIMIT
 8. MAKE (pMonitoredValue a value that is between pLowLimit and pHighLimit - pDeadband)
 9. WAIT (pTimeDelayNormal)
 10. BEFORE Notification Fail Time

RECEIVE ConfirmedEventNotification-Request,	
'Process Identifier' =	(any valid process ID),
'Initiating Device Identifier' =	IUT,
'Event Object Identifier' =	(the object configured for this test),
'Time Stamp' =	(the current local time),
'Notification Class' =	(the class corresponding to the object being tested),
'Priority' =	(the value configured to correspond to a TO-NORMAL transition),
'Event Type' =	<i>(the algorithm appropriate to the object type i.e. OUT_OF_RANGE, SIGNED_OUT_OF_RANGE, UNSIGNED_OUT_OF_RANGE, or DOUBLE_OUT_OF_RANGE),</i>

'Message Text' =	(optional, any valid message text),
'Notify Type' =	ALARM EVENT,
'AckRequired' =	TRUE FALSE,
'From State' =	HIGH_LIMIT,
'To State' =	NORMAL,
'Event Values' =	(values appropriate to the event type)
 11. TRANSMIT SimpleAck-PDU
 12. MAKE pLimitEnable = (?, FALSE)
 13. VERIFY pCurrentState = NORMAL
 14. MAKE (pMonitoredValue a value that exceeds pHighLimit)
 15. WAIT (pTimeDelay + Notification Fail Time)
 16. CHECK (verify that no notification message was transmitted)
 17. VERIFY pCurrentState = NORMAL

Notes to Tester: The UnconfirmedEventNotification service may be substituted for the ConfirmedEventNotification service in which case the TD shall skip all of the steps in which a SimpleACK-PDU is sent.

BTL-15.2o-5: Clarify the Expected Behavior for COV Notifications (BTLWG-402)

Overview:

The current tests for COV do not prohibit devices from sending unlimited COVs. There has been debate about how many COVs are permitted when considering changes that may also occur to the status_flags that are the result of a change to Present_Value. The work item attempts to clarify the expected behavior regarding the generation of COV notifications.

Changes:

[In BTL Specified Tests, add the following notes to tester to COV tests (Present_Value version only)

BTL - 8.2.1 - Change of Value Notification from an Analog Input, Analog Output, and Analog Value, *Large Analog Value, Integer Value, and Positive Integer Value* Object Present_Value Property

BTL - 8.3.1 - Change of Value Notification from an Analog Input, Analog Output, and Analog Value, *Large Analog Value, Integer Value, and Positive Integer Value* Object Present_Value Property

BTL - 8.2.3 - Change of Value Notification from a Binary Input, Binary Output, and Binary Value Object Present_Value Property

BTL - 8.2.5 - Change of Value Notification from a Multi-state Input, Multi-state Output, Multi-state Value, Life Safety Point, and Life Safety Zone, *CharacterString Value, OctetString Value, Date Value, Date Pattern Value, DateTime Value, DateTime Pattern Value, Time Value, or Time Pattern Value* Object Present_Value Property

BTL - 8.3.5 - Change of Value Notification from a Multi-state Input, Multi-state Output, Multi-state Value, Life Safety Point, and Life Safety Zone, *CharacterString Value, OctetString Value, Date Value, Date Pattern Value, DateTime Value, BTL - 8.2.7 - Change of Value Notification from Loop Object Present_Value Property* *DateTime Pattern Value, Time Value, or Time Pattern Value* Object Present_Value Property

BTL - 8.2.X9 - ConfirmedCOVNotification Pulse Converter changing Present_Value

BTL - 8.3.X12 - UnconfirmedCOVNotification Pulse Converter changing Present_Value

]

Notes to tester: The IUT may initiate additional COVNotifications. The final COVNotification shall accurately reflect Present_Value and Status_Flags.

[In BTL Specified Tests, copy the following tests from 135.1-2013 into BTL Specified Tests and then add the following notes to tester and Reason for Change.

135.1-2013 - 8.3.3 - Change of Value Notification from a Binary Input, Binary Output, and Binary Value Object Present_Value Property

135.1-2013 - 8.3.7 - Change of Value Notification from Loop Object Present_Value Property

]

Reason for Change: Add clarification to test that the last COVNotification shall reflect the correct values.

Notes to tester: The IUT may initiate additional COVNotifications. The final COVNotification shall accurately reflect Present_Value and Status_Flags.

[In BTL Test Plan, modify any reference for 135.1-2013 - 8.3.3 to be BTL - 8.3.3]

[In BTL Test Plan, modify any reference for 135.1-2013 - 8.3.7 to be BTL - 8.3.7]

BTL-15.2o-6: Add Array Resizing Test for WPM Service (BTLWG-444)

Overview:

Add new test for resizing of an array using WritePropertyMultiple Service.

Changes:

[Change in Test Plan section 4.8, Data Sharing - WritePropertyMultiple - B]

4.8 Data Sharing-WritePropertyMultiple - B

4.8.7 Contains Resizable Array Properties

The IUT contains, or can be made to contain, an array property that is resizable by writing to the 0th element.

...	
135.1 2013 7.3.1.16 Array Sizing Test <i>BTL - 7.3.1.X16 - Array Resizing Test using WritePropertyMultiple Service</i>	
Test Conditionality	This test shall be executed if the IUT is protocol revision 4 or higher and contains an object with a resizable BACnetARRAY property.
Test Directives	<i>Execute on at least one instance of each resizable array property, both standard and proprietary</i>
Testing Hints	Use WritePropertyMultiple instead of WriteProperty for this instance of the test.

[Add this new test in BTL Specified Tests, as shown]

7.3.1.X16 Array Resizing Test using WritePropertyMultiple Service

Purpose: To verify that resizable arrays are resized in accordance with the rules added in Protocol_Revision 4.

Test Concept: The resizable array property P1 of object O1 is written with WritePropertyMultiple as a whole to set it to a non-zero size. It is then resized smaller and larger by writing the entire array. It is then resized smaller and larger by writing to element number zero. An attempt is made to increase it with an invalid write. After each operation, the array size and array contents are checked. Finally, if it can be resized to have zero elements, it is then written to size zero. If possible, all elements in the arrays should be distinguishable from each other and across WritePropertyMultiple operations.

Test Steps:

1. TRANSMIT WritePropertyMultiple-Request,
 - 'Object Identifier' = O1,
 - 'Property Identifier' = P1,
 - 'Property Value' = (array A1 of non-zero size N1)
2. RECEIVE BACnet-SimpleACK-PDU
3. VERIFY P1= (array A1), ARRAY INDEX = 0, (array size i.e. N1)

--Resize the array to make it smaller in size

4. TRANSMIT WritePropertyMultiple-Request,
 - 'Object Identifier' = O1,
 - 'Property Identifier' = P1
 - 'Property Value' = (array A2 of non-zero size N2, where $N2 \leq N1$)
5. RECEIVE BACnet-SimpleACK-PDU
6. VERIFY P1 = (array A2), ARRAY INDEX = 0, (array size N2)

--Resize the array to make it larger in size

7. TRANSMIT WritePropertyMultiple-Request,
 - 'Object Identifier' = O1,
 - 'Property Identifier' = P1
 - 'Property Value' = (array A3 of non-zero size N3, where $N3 \geq N1$),
8. RECEIVE BACnet-SimpleACK-PDU

9. VERIFY P1 = (array A3), ARRAY INDEX = 0, (array size N3)

--Modify the existing content of element

10. TRANSMIT WritePropertyMultiple-Request,
 'Object Identifier' = O1,
 'Property Identifier' = P1,
 'Property Value' = (array A4 of non-zero unsigned value N4, where $N4 \leq N1$),

11. RECEIVE BACnet-SimpleACK-PDU

12. VERIFY P= (array A4), ARRAY INDEX = 0, (array size N4)

--Resize the array by writing the size of the array

13. TRANSMIT WritePropertyMultiple-Request
 'Object Identifier' = O1,
 'Property Identifier' = P1
 'Property Value' = (N5, where $N5 \geq N4$),
 'Property Array Index' = 0,

14. RECEIVE BACnet-SimpleACK-PDU

15. VERIFY (array contains unchanged first N4 elements of the array written in step 10, plus N5-N4 additional elements, initialized to particular values for the array property being tested)

16. VERIFY P1, ARRAY INDEX = 0, (array size N5)

--Try to add the array element at Array Index which is greater than the size of the array

17. TRANSMIT WritePropertyMultiple-Request,
 'Object Identifier' = O1,
 'Property Identifier' = P1,
 'Property Value' = (one array element),
 'Property Array Index' = (N6, where $N6 \geq N5$),

18. RECEIVE WritePropertyMultiple-Error
 'Error Class' = PROPERTY,
 'Error Code' = INVALID_ARRAY_INDEX
 'Object Identifier' = O1,
 'Property Identifier' = P1,
 'Property Array Index' = N6
 |
 'Error Class' = RESOURCES,
 'Error Code' = NO_SPACE_TO_WRITE_PROPERTY
 'Object Identifier' = O1,
 'Property Identifier' = P1
 'Property Array Index' = N6

19. VERIFY (array is unchanged from step 15)

--Resize the array to size zero

20. IF (the array can be resized to have zero elements) THEN
 TRANSMIT WritePropertyMultiple-Request,
 'Object Identifier' = O1,
 'Property Identifier' = P1,
 'Property Value' = (empty array)
 RECEIVE BACnet-SimpleACK-PDU
21. VERIFY P1 = (array is empty), ARRAY INDEX = 0, (array size is zero)

BTL-15.20-7: Remove Section 9.4.7 Is Able to Register as Foreign Device in BBMD (BTLWG-550)

Overview:

Registering as a Foreign Device while acting as a BBMD is an invalid configuration.

Changes:

[In the test plan, strike 9.4.7 entirely and remove Distribute Broadcast To Network from 9.4.8]

9 Data Link Layer

9.4 BACnet/IP - Annex J - BBMD

~~9.4.7 Is Able to Register as a Foreign Device~~

While configured as a BBMD, the IUT is able to register as a Foreign Device with another BBMD.

135.1 2013 14.8 Registering as a Foreign Device		
	Test Method	Manual
	Configuration	As per ASHRAE 135.1-2013.
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
	Notes & Results	
135.1 2013 14.9.1 Distribute Broadcast To Network		
	Test Method	Manual
	Configuration	As per ASHRAE 135.1-2013.
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
	Notes & Results	

~~9.4.8 9.4.7 Is Able to Initiate Original-Broadcast-NPDU~~

While configured as a BBMD, the IUT can issue broadcasts on its own local subnet and through partner BBMDs.

135.1 2013 14.9.1 Distribute Broadcast To Network		
	Test Method	Manual
	Configuration	As per ASHRAE 135.1-2013.
	Test Conditionality	This test shall be skipped if the IUT cannot register as a Foreign device while acting as a BBMD.
	Test Directives	
	Testing Hints	
	Notes & Results	
135.1-2013 - 14.9.3 - Original-Broadcast-NPDU		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

[In the checklist, strike the corresponding test item]

Data Link Layer

Support	Listing	Option
...		
BACnet/IP - Annex J - BBMD		
	R	Base Requirements
	BTL-R	Supports a BDT with at least four entries
	R	Registration by a Foreign Device is supported
	R	Supports 2-hop mode
	O	Supports 1-hop mode
	O	BBMD supports Network Address Translation
	⊖	Is able to register as a Foreign Device
	O	Is able to initiate Original-Broadcast-NPDU
...		

BTL-15.2o-8: Remove Hop-Count Protection from Virtual Tests (BTLWG-271)

Overview:

In a router only to virtual networks, that router won't ever route a message incoming to the Virtual Device back out again, so that could never form a loop, so needs no Hop Count checking behavior implemented in the IUT. Test 10.8.5 is thus not applicable to a router to a virtual network and should be removed from the test package.

Changes:

[In the BTL Test Plan, strike a test, so that testing is not required to be executed if a Virtual Router.]

10.1.3 Routes Packets Between a Physical LAN and One or More Virtual LANs

The device can route BACnet packets between a physical BACnet LAN and one or more virtual BACnet LANs that contain one or more virtual BACnet devices. See H.1 and H.2 in the BACnet standard for a description of virtual BACnet LANs and virtual BACnet devices.

...		
135.1 2013 10.8.5 Hop Count Protection		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

BTL-15.2o-9: Modify Pulse Converter Test to Correctly Indicate How to Change Status_Flags (BTLWG-461)

Overview:

The test specification in BTL TP 15.2 tries to write to the Status_Flags property, which is not considered a normal operation.

Changes:

[In BTL Specified Tests, modify test 8.2.X10 by adding the information in yellow.]

8.2.X10 ConfirmedCOVNotification Pulse Converter changing Status_Flags

Purpose: To verify the correct operation of COV in the Pulse Converter object. The Pulse Converter initiates periodic COV Notifications every COV_Period, even when there are no changes in the object, in addition to the COV notifications that this object type generates due to changes in the Status_Flags property.

Test Concept: A subscription for COV notifications is established, using a Lifetime of L. L shall be set to a value less than 8 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. The object identifier of the Pulse Converter object being tested is designated as O1 in the test steps below.

Configuration Requirements: At the beginning of the test, the Out_Of_Service property shall have a value of FALSE. Select an object where Present_Value is not expected to change outside the tester's control by more than COV_Increment ~~or which has a writable Out_Of_Service.~~ COV_Period is configured high enough that it does not trigger many COV notifications during the execution of the test.

Notes to Tester: Additional COV notifications from O1 can occur after step 3 due to COV_Period cyclic notifications.

Test Steps:

1. TRANSMIT SubscribeCOV-Request,
 - 'Subscriber Process Identifier' = (any value > 0 chosen by the TD),
 - 'Monitored Object Identifier' = O1,
 - 'Issue Confirmed Notifications' = TRUE,
 - 'Lifetime' = L
2. RECEIVE BACnet-SimpleACK-PDU
3. BEFORE **Notification Fail Time**
 - RECEIVE ConfirmedCOVNotification-Request,
 - 'Subscriber Process Identifier' = (the same value used in step 1),
 - 'Initiating Device Identifier' = IUT,
 - 'Monitored Object Identifier' = O1,
 - 'Time Remaining' = (any value appropriate for the Lifetime selected),
 - 'List of Values' = (the initial Present_Value, initial Status_Flags, and Update_Time)
4. TRANSMIT BACnet-SimpleACK-PDU
5. IF (Out_Of_Service is writable) THEN
 - WRITE Out_Of_Service = TRUE
 - ELSE
 - WRITE O1, Out_Of_Service = TRUE | WRITE O1, Status_Flags=(a value that differs from initial Status_Flags)

MAKE Status_Flags = any value that differs from initial Status_Flags)
6. BEFORE **Notification Fail Time**
 - RECEIVE ConfirmedCOVNotification-Request,
 - 'Subscriber Process Identifier' = (the same value used in step 1),
 - 'Initiating Device Identifier' = IUT,
 - 'Monitored Object Identifier' = O1,
 - 'Time Remaining' = (any value appropriate for the Lifetime selected),
 - 'List of Values' = (the current Present_Value, new Status_Flags, and Update_Time)
7. TRANSMIT BACnet-SimpleACK-PDU
8. TRANSMIT SubscribeCOV-Request,

'Subscriber Process Identifier' = (the same value used in step 1),
'Monitored Object Identifier' = O1

9. RECEIVE BACnet-SimpleACK-PDU

10. IF (Out_Of_Service is-writable ~~was changed in step 5~~) THEN
WRITE ~~O1~~, Out_Of_Service = FALSE

BTL-15.2o-10: Move COV-A Negative Tests (BTLWG-54)

Overview:

CR-0408 proposed that the COV-A negative tests, which can only be executed if ConfirmedCOV is supported, should be moved from the Base Req to the section for ConfirmedCOV.

Changes:

[In BTL Test Plan 15.2, move the 3 entries from the COV-A “Base Requirements” to “Can Subscribe for Confirmed Notifications” and remove the test conditionality]

4.9.1 Base Requirements

Base requirements must be met by any IUT claiming conformance to this BIBB. *There are no base requirements for this section.*

BTL - 9.2.2.1 - Change of Value Notification Arrives after Subscription has Expired	
Test Method	
Configuration	As per <i>BTL Specified Tests</i> .
Test Conditionality	Must be executed if IUT supports initiation of SubscribeCOV Request with ‘Issue ConfirmedNotifications’ equal to TRUE.
Test Directives	
Testing Hints	
Notes & Results	
BTL - 9.2.2.2 - Change of Value Notifications with Invalid Process Identifier	
Test Method	
Configuration	As per <i>BTL Specified Tests</i> .
Test Conditionality	Must be executed if IUT supports initiation of SubscribeCOV Request with ‘Issue ConfirmedNotifications’ equal to TRUE.
Test Directives	
Testing Hints	
Notes & Results	
BTL - 9.2.2.4 - Change of Value Notifications with Invalid Monitored Object Identifier	
Test Method	
Configuration	As per <i>BTL Specified Tests</i> .
Test Conditionality	Must be executed if IUT supports initiation of SubscribeCOV Request with ‘Issue ConfirmedNotifications’ equal to TRUE.
Test Directives	
Testing Hints	
Notes & Results	

4.9.3 Can Subscribe for Confirmed Notifications

The IUT can subscribe for, receive, and process confirmed Change of Value notifications.

135.1-2013 - 8.10.1 - Confirmed Notifications Subscription	
Test Conditionality	Must be executed.
Test Directives	
Testing Hints	
<i>BTL - 9.2.2.1 - Change of Value Notification Arrives after Subscription has Expired</i>	
Test Conditionality	<i>Must be executed.</i>

	<i>Test Directives</i>	
	<i>Testing Hints</i>	
<i>BTL - 9.2.2.2 - Change of Value Notifications with Invalid Process Identifier</i>		
	<i>Test Conditionality</i>	<i>Must be executed.</i>
	<i>Test Directives</i>	
	<i>Testing Hints</i>	
<i>BTL - 9.2.2.4 - Change of Value Notifications with Invalid Monitored Object Identifier</i>		
	<i>Test Conditionality</i>	<i>Must be executed.</i>
	<i>Test Directives</i>	
	<i>Testing Hints</i>	

BTL-15.2o-11: Clarify Internally Written Datatypes Test (BTLWG-274)

Overview:

CR-0315 Response:

In the tests where a write is specified but the Schedule cannot be made to reference any property, the effect should be observed in the Present_Value property of the Schedule Object. The tests, Test Plan section preambles, and Test Conditionality wording will be adjusted in a future Test Package.

Changes:

[In BTL Test Plan, change all test references for 135.1-2013 - 7.3.2.23.11.1 to BTL - 7.3.2.23.11.1, there are 38]

[In BTL Specified Tests, derive from existing tests in 135.1-2013, changing the Configuration Requirements so that the test can be conducted, and effects can be observed in the Present_Value property of the Schedule Object, when there is no Schedule that can be made to reference another property.]

7.3.2.23.11.1 Internally Written Datatypes Test, non-NULL values

Reason for Change: To ensure that datatype-specific testing is conducted even when there is no Schedule that can be made to reference another property. The effect should be observed in the Present_Value property of the Schedule Object instead.

BACnet Reference Clauses: 12.24, 12.24.10

Purpose: This test verifies that the Schedule object within the IUT writes to properties in the same device for the non-NULL datatype being tested.

Test Concept: Two Date/Time, values, D1 and D2, are chosen by the TD based on the criteria in Table 7-17 such that D1 is sufficiently different from, and later than, the current time to cause a Schedule evaluation when the time is changed to D1, and such that setting the time to D2 (later than D1) from D1 will cause a Schedule evaluation that will cause it to write value V2. These values may be chosen based on the Schedule object’s existing configuration, or the Schedule object, S, may be configured with such values.

Configuration Requirements: The IUT shall be configured with a Schedule object, S, such that the time periods defined in Table 7-17 can be configured with uniquely scheduled values. The Schedule object shall be configured with a List Of Object Property References, including at least one reference to a writable property within the device, *if possible. Step 4 and step 8 would REPEAT zero times, if the referenced property is empty or not present. If the IUT cannot be configured to these requirements, then this test shall be omitted. Other pProperties in the Schedule object* shall be consistent in both datatypes and values in a manner permitting this test to be executed.

Table 7-17. Criteria for Test Date and Times

Date and Time:	Value:
D ₁	V ₁
D ₂	V ₂ different from V ₁ .

Test Steps:

1. (TRANSMIT TimeSynchronization-Request, 'Time' = D₁)
 - | (TRANSMIT UTCTimeSynchronization-Request, 'Time'=D₁)
 - | MAKE (the local date and time = D₁)
2. WAIT (**Schedule Evaluation Fail Time**)
3. VERIFY S, Present_Value = V₁
4. REPEAT P = (writable property in List_Of_Property_References)
 - VERIFY P = V₁
5. (TRANSMIT TimeSynchronization-Request, 'Time' = D₂)
 - | (TRANSMIT UTCTimeSynchronization-Request, 'Time'=D₂)
 - | MAKE (the local date and time = D₂)

6. **WAIT(Schedule Evaluation Fail Time)**
7. VERIFY S, Present_Value = V_2
8. REPEAT P = (writable property in List_Of_Property_References)
 VERIFY P = V_2

Note to Tester: In the context of this test definition, writable means that the Schedule object is capable of modifying the property. It does not necessarily indicate that the property is modifiable via BACnet services.

BTL-15.2o-12: Update Device Restart Test for Sequence Numbers (BTLWG-459)

Overview:

CR-0437 pointed out that test 135.1-2013 8.3.10 does not work for devices that don't have a Local_Time property and use a sequence number in Restart Notifications.

Changes:

[Update in the Test Plan to reference this test in BTL Specified Tests, rather than 135.1-2013]

[In BTL Specified Tests, add test 8.3.10 Device Restart Notifications from 135.1 and modify as shown below]

8.3.10 Device Restart Notifications

Reason for Change: CR-0437 pointed out that test 135.1-2013 8.3.10 does not work for devices that don't have a Local_Time property and use a sequence number in Restart Notifications.

Purpose: To verify that the IUT initiates UnconfirmedCOVNotification service requests to each entry in its Restart_Notification_Recipients property when it resets.

Test Concept: The IUT is configured to send restart notifications and is then reset. The TD checks for the restart notifications.

Device restart notifications differ from subscribed COV notifications that use the UnconfirmedCOVNotification service in two respects. First, subscription is made through the Restart_Notification_Recipients property instead of SubscribeCOV. Second, the 'Subscriber Process Identifier' parameter always has a value of zero.

Configuration Requirements: For each Recipient of the Restart_Notification_Recipients property in the IUT which is of the device form, there shall be a device on the network that will answer Who-Is requests so that the IUT can determine addressing information before sending the restart notification.

Test Steps:

1. IF (Restart_Notification_Recipients is writable) THEN
 WRITE(Restart_Notification_Recipients = any non-empty list of Recipients)
 ELSE
 MAKE (Restart_Notification_Recipients contain any non-empty list of Recipients)
- ~~2. READ T1 = Local_Time~~
3. MAKE(the IUT reset)
4. REPEAT X = (each entry in the Restart_Notification_Recipients) DO {
 BEFORE Notification Fail Time
 RECEIVE UnconfirmedCOVNotification-Request,
 DESTINATION = X,
 'Subscriber Process Identifier' = 0,
 'Initiating Device Identifier' = IUT,
 'Monitored Object Identifier' = (the IUT Device Identifier),
 'Time Remaining' = 0,
 'List of Values' = (System_Status=OPERATIONAL,
 Time_Of_Device_Restart = (T2),
 Last_Restart_Reason=(any valid restart reason, R))
 }
5. VERIFY Time_Of_Device_Restart = T2
- ~~6. CHECK (T1 ~ T2)~~
7. VERIFY Last_Restart_Reason = R
6. IF (T2 is not a sequence number) THEN
 VERIFY Local_Time ~ T2
 ELSE
 MAKE(the IUT reset)
 REPEAT X = (each entry in the Restart_Notification_Recipients) DO {

BEFORE Notification Fail Time

RECEIVE UnconfirmedCOVNotification-Request,

DESTINATION = X,

'Subscriber Process Identifier' = 0,

'Initiating Device Identifier' = IUT,

'Monitored Object Identifier' = (the IUT Device Identifier),

'Time Remaining' = 0,

'List of Values' = (System_Status=OPERATIONAL,

Time Of Device Restart = (T3),

Last Restart Reason=(any valid restart reason, R))

CHECK (T3 > T2)

Note to tester: Not all IUTs can accurately differentiate between the types of restart reasons and thus no requirements are placed on the value returned in *the restart notification(s)* step 4. The test shall pass regardless of the order in which the *restart notifications are sent to the recipients*. IUT generates the UnconfirmedCOVNotification-Requests in step 4. The value of T2 shall be the same for each notification sent out in step 4. *If the Restart_Notification_Recipients list has multiple recipients, then the Time_Of_Device_Restart value is expected to be the same in all notifications resulting from the same restart.*

BTL-15.2o-13: Update Schedule Test Directives for Single Exception Schedule Devices (BTLWG-598)

Overview:

CR-0432 pointed out that schedules need only support 1 exception schedule entry, but some tests require 2. The test conditionality is changed for these tests.

Changes:

[In BTL Test Plan 15.2, section 6.4.2, change the conditionality of BTL - 7.3.2.23.10.3.8 and 135.1-2013 - 7.3.2.23.10.3.12]

BTL - 7.3.2.23.10.3.8 - Revision 4 Event Priority Test	
Test Conditionality	As per <i>BTL Specified Tests</i> if and only If the IUT is protocol revision 4 or higher. If the IUT is of the correct Protocol_Revision, the IUT is required to be configurable such that this test can be run. This test may not be skipped. <i>If the IUT does not support enough exception schedule entries to execute this test, the test shall be skipped, otherwise the test shall be executed.</i>
Test Directives	
Testing Hints	

135.1-2013 - 7.3.2.23.10.3.12 - Revision 4 Lower Event Priority Change Test	
Test Conditionality	As per <i>BTL Specified Tests</i> if and only If the IUT is protocol revision 4 or higher. If the IUT is of the correct Protocol_Revision, the IUT is required to be configurable such that this test can be run. This test may not be skipped. <i>If the IUT does not support enough exception schedule entries to execute this test, the test shall be skipped, otherwise the test shall be executed.</i>
Test Directives	
Testing Hints	