



BACnet<sup>®</sup> TESTING LABORATORIES

# INTERIM TEST SPECIFICATION

To Be Used with Test Package 18.1  
Version 2  
July 15, 2021

Approved by the BTL Working Group on June 10, 2021.  
Approved by the BTL Working Group Voting Members on July 6, 2021.  
Published on July 16, 2021.

## Foreward

The purpose of this document is to define interim tests and other test package changes made to support testing of a device that supports functionality currently not covered in the released BTL Test Package. This document shall be applied and used with BTL Test Package 18.1.

Vendors who are planning to submit a device for testing and who implement Protocol\_Revision 19 and higher, or which contain functionality not covered by the Official Test Package, should use this Interim Test document.

Please note that if the device contains functionality not yet covered by the official Test Package, nor by the Interim Tests document, development of new tests may be required for your device. Please contact the BTL Manager before submitting your device for testing to ensure you are aware of all tests that will need to be applied to your device.

The changes in this document are for interim use only and may or may not be used as documented here when the final changes are applied to the next Test Package revision. Devices tested using this interim test document shall be recalled for updated testing when the next revision of test package is released that includes the topics covered here.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135.1-2019 or any part of the Test Package 18.1 are indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new sections are proposed to be added, plain type is used throughout.

## Table of Contents

<b>BTL CHECKLIST AND BTL TEST PLAN CHANGES .....</b>	<b>3</b>
5.5 Alarm and Event Management - Acknowledgement - B .....	4
13.5 Audit Reporting-View-A.....	5
<b>BTL SPECIFIED TESTS CHANGES.....</b>	<b>8</b>
7.3.1.11 Acked_Transitions Tests.....	9
7.3.1.11.1 Acked_Transitions Tests Test.....	9
7.3.1.11.2 Acked_Transitions Test for Latching Objects.....	9
8.X AuditLogQuery Initiation Tests.....	11
8.X.1 Query and Present Audit Log Records By Source .....	11
8.X.2 Query and Present Audit Log Records By Target.....	11

---

## **BTL Checklist and BTL Test Plan Changes**

---

This section of the document contains interim changes to the BTL Checklist and the BTL Test Plan documents to support testing of products with functionality outside the scope of the official test plan.

This section is ordered the same as the BTL Checklist and BTL Test Plan documents to allow easy navigation of the material.

All test changes can be found in the next major section.

## 5.5 Alarm and Event Management - Acknowledgement - B

The current Test Package does not properly take into account latching life safety objects. These changes address that insufficiency.

### Checklist Changes

None

### Test Plan Changes

[Modify the 7.3.1.11 entry in AE-ACK-B Base Requirements, section 5.5.1]

<b>BTL - 7.3.1.11 - Acked Transitions Tests</b>	
<b>BTL - 7.3.1.11.1 - Acked Transitions Test</b>	
<b>Test Conditionality</b>	<del>Must be executed.</del> <i>If the IUT only supports event generating objects which latch their event state and require an acknowledgement before unlatching, this test shall be skipped.</i> <i>Only life safety objects are allowed to latch in this manner.</i>
<b>Test Directives</b>	
<b>Testing Hints</b>	

[Add an entry for 7.3.1.11.2 immediately after the entry for 7.3.1.11.1 in AE-ACK-B Base Requirements, section 5.5.1]

<b>BTL - 7.3.1.11.2 - Acked Transitions Test for Latching Objects</b>	
<b>Test Conditionality</b>	If the IUT does not support event generating objects which latch their event state and require an acknowledgement before unlatching, this test shall be skipped. Only life safety objects are allowed to latch in this manner.
<b>Test Directives</b>	Apply the test for each supported transition.
<b>Testing Hints</b>	

## 13.5 Audit Reporting-View-A

Addendum 135-2016*bi* added Audit Reporting. This section adds support to the BTL Test Package for claiming AR-V-A and AR-AVM-A.

These changes are not contained in any SSPC proposal.

### Checklist Changes

[In BTL Checklist, replace Audit Reporting sections]

Audit Reporting - View - A		
	R <sup>+</sup>	Base Requirements
	C <sup>1</sup>	Supports initiation of AuditLogQuery by Target
	C <sup>1</sup>	Supports initiation of AuditLogQuery by Source
	C <sup>1</sup>	Supports initiation of ReadRange
<sup>1</sup> At least one of these must be supported.		
Audit Reporting - Advanced View and Modify - A		
	R <sup>+</sup>	Base Requirements
	C <sup>1</sup>	Supports initiation of AuditLogQuery by Target
	C <sup>1</sup>	Supports initiation of AuditLogQuery by Source
<sup>1</sup> At least one of these must be supported.		

### Test Plan Changes

[In BTL Test Plan, replace section 13.5 Audit Reporting-View-A]

## 13.5 Audit Reporting-View-A

### 13.5.1 Base Requirements

Base requirements must be met by any IUT that supports AR-V-A.

### 13.5.2 Supports Initiation of AuditLogQuery By Target

BTL - 8.X.2 - Query and Present Audit Log Records By Target		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

### 13.5.3 Supports Initiation of AuditLogQuery By Source

BTL - 8.X.1 - Query and Present Audit Log Records By Source		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

### 13.5.4 Supports Initiation of ReadRange

BTL - 8.18.X1 - Reading and Presenting Large List Properties		
	Test Conditionality	Must be executed.
	Test Directives	Apply on Log_Buffer property of an AuditLog and verify that each record is completely presented.
	Testing Hints	

---

## 13.6 Audit Reporting-Advanced View and Modify-A

---

### 13.6.1 Base Requirements

Base requirements must be met by any IUT that supports AR-AVM-A.

<b>Verify Checklist</b>		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims AR-V-A
	Testing Hints	
<b>Verify Checklist</b>		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims DS-RP-A
	Testing Hints	
<b>Verify Checklist</b>		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims DS-WP-A
	Testing Hints	
<b>Verify Checklist</b>		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims DM-OCD-A and is able to create Audit Reporter and Audit Log objects.
	Testing Hints	
<b>135.1-2019 - 8.18.3 - Reading and Presenting Properties</b>		
	Test Conditionality	Must be executed.
	Test Directives	Repeat for all properties of the Audit Reporter and Audit Log objects specified by AR-AVM-A, and for all audit related properties in a randomly chosen set of other standard object types.
	Testing Hints	
<b>135.1-2019 - 8.22.4 - Accepting Input and Modifying Properties</b>		
	Test Conditionality	Must be executed.
	Test Directives	Repeat for all properties of the Audit Reporter and Audit Log objects specified by AR-AVM-A, and for all audit related properties in a randomly chosen set of other standard object types.
	Testing Hints	

### 13.6.2 Supports Initiation of AuditLogQuery By Target

<b>Verify Checklist</b>		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims support for AuditLogQuery by Target for AR-V-A.
	Testing Hints	

### 13.6.3 Supports Initiation of AuditLogQuery By Source

Verify Checklist		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Verify that the IUT claims support for AuditLogQuery by Source for AR-V-A.
	<b>Testing Hints</b>	



---

## **BTL Specified Tests Changes**

---

This section contains all of the new and changed tests required by the interim test BTL Checklist and BTL Test Plan changes.

[Latching lifesafety object test changes]

[Add new clause 7.3.1.11]

### 7.3.1.11 Acked\_Transitions Tests

[Renumber and rename test 7.3.1.11 Acked\_Transitions Tests to 7.3.1.11.1 Acked\_Transitions Test]

#### 7.3.1.11.1 Acked\_Transitions Tests Test

[Add a new test into BTL Specified Tests]

#### 7.3.1.11.2 Acked\_Transitions Test for Latching Objects

Reason for Change: No test exists for this functionality.

Purpose: To verify that the Acked\_Transitions property tracks the acknowledgment state for a transition type.

Test Concept: This test is a single transition test for latching life safety objects which are not able to perform the regular Acked\_Transitions test for all transitions. An object, O1, in the IUT is made to generate a transition which requires an acknowledgement. The Acked\_Transitions property is verified that the corresponding flag is cleared (set to FALSE). The transition is acknowledged, and the flag is verified to have been set back to TRUE.

Configuration Requirements: O1 is configured to generate events and to require acknowledgements for the transition being tested. O1 should have no event transitions which have outstanding acknowledgements.

Test Steps:

1. VERIFY Acked\_Transitions = (TRUE, TRUE, TRUE)
2. MAKE (O1 transition)
3. WAIT (pTimeDelay)
4. BEFORE **Notification Fail Time**
  - RECEIVE ConfirmedEventNotification-Request,
    - 'Process Identifier' = (PI1: any valid process ID),
    - 'Initiating Device Identifier' = IUT,
    - 'Event Object Identifier' = O1,
    - 'Time Stamp' = (T1: any valid time stamp),
    - 'Notification Class' = (NC1: the class corresponding to the object being tested),
    - 'Priority' = (PRIO1: the value configured to correspond to the transition type),
    - 'Event Type' = (E1: any valid event type),
    - 'Message Text' = (optional, any valid message text),
    - 'Notify Type' = (the notify type configured for this event),
    - 'AckRequired' = TRUE,
    - 'From State' = S1,
    - 'To State' = S2,
    - 'Event Values' = (values appropriate to the event type)
5. TRANSMIT BACnet-SimpleACK-PDU
6. VERIFY pCurrentState = S2
7. IF S2 is NORMAL THEN
  - VERIFY Acked\_Transitions = (TRUE, TRUE, FALSE)
  - VERIFY pStatusFlags = (FALSE, FALSE, ?, ?)ELSE IF S2 is FAULT THEN
  - VERIFY Acked\_Transitions = (TRUE, FALSE, TRUE)
  - VERIFY pStatusFlags = (TRUE, TRUE, ?, ?)ELSE
  - VERIFY Acked\_Transitions = (FALSE, TRUE, TRUE)
  - VERIFY pStatusFlags = (TRUE, FALSE, ?, ?)
8. TRANSMIT AcknowledgeAlarm-Request,
  - 'Acknowledging Process Identifier' = PI1,

```

    'Event Object Identifier' =      O1,
    'Event State Acknowledged' =    S2,
    'Time Stamp' =                  T1,
    'Acknowledgement Source' =      (a character string),
    'Time of Acknowledgment' =      (the TD's current time)
9. RECEIVE BACnet-SimpleACK-PDU
10. IF (Protocol_Revision is present and Protocol_Revision ≥ 1) THEN
    BEFORE Notification Fail Time
        RECEIVE ConfirmedEventNotification-Request,
            'Process Identifier' =    PI1,
            'Initiating Device Identifier' = IUT,
            'Event Object Identifier' = O1,
            'Time Stamp' =            T1,
            'Notification Class' =    NC1,
            'Priority' =               PRIO1,
            'Event Type' =            E1,
            'Message Text' =          (optional, any valid message text),
            'Notify Type' =           ACK_NOTIFICATION,
            'To State' =              S2
    ELSE
        BEFORE Notification Fail Time
            RECEIVE ConfirmedEventNotification-Request,
                'Process Identifier' =    PI1,
                'Initiating Device Identifier' = IUT,
                'Event Object Identifier' = O1,
                'Time Stamp' =            T1,
                'Notification Class' =    NC1,
                'Priority' =               PRIO1,
                'Event Type' =            E1,
                'Message Text' =          (optional, any valid message text),
                'Notify Type' =           ACK_NOTIFICATION
11. TRANSMIT BACnet-SimpleACK-PDU
12. VERIFY Acked_Transitions = (TRUE, TRUE, TRUE)

```

Notes to Tester: The UnconfirmedEventNotification service may be substituted for the ConfirmedEventNotification service, in which case the TD shall skip sending the BACnet-SimpleACK-PDU messages after receiving the notifications.

[ Audit Reporting Tests ]  
[ Insert clause 8.X ]

## 8.X AuditLogQuery Initiation Tests

This clause defines the tests necessary to demonstrate support for initiating AuditLogQuery service requests.

### 8.X.1 Query and Present Audit Log Records By Source

Reason for Change: no tests exist for the functionality.

Purpose: To verify that the IUT correctly initiates AuditLogQuery requests and presents the results.

Test Concept: TD is setup as an audit logger with an Audit Log. The IUT is made to request and display the contents of the Audit Log for source S1. The results are verified that they match the content of the log.

Test Configuration: The TD is setup with an AuditLog containing content from multiple sources and for multiple targets. The audit log contains example entries of all possible operations (see clause 19.Y.5) for audit source S1 with a mix of success and failure entries

Test Steps:

1. WHILE (the IUT has not retrieved and displayed all entries for S1)  
    MAKE (the IUT request more content from the Audit Log)  
    RECEIVE AuditLogQuery-Request  
        'Audit Log' = (the audit log in the TD),  
        'Query Parameters' = (a valid Audit Query by Source query including S1 as  
                                  the source),  
        'Start At Sequence Number' = (any valid value)  
        'Requested Count' = (any valid value)  
    TRANSMIT AuditLogQuery-Result  
        'Audit Log' = (the audit log in the TD),  
        'Records' = (the set of audit log records which match the query  
                      and which fit within the accepted response size),  
        'No More Items' = (TRUE if the last item is included, FALSE otherwise)
2. CHECK(that the displayed content matches audit records returned and that the complete records are  
    presented)

Notes to Tester: If manual interaction is required between subsequent AuditLogQuery requests, checking of the displayed content might need to be performed before the manual interaction is taken instead of at the end of retrieving all of the items.

### 8.X.2 Query and Present Audit Log Records By Target

Reason for Change: no tests exist for the functionality.

Purpose: To verify that the IUT correctly initiates AuditLogQuery requests and presents the results.

Test Concept: TD is setup as an audit logger with an Audit Log. The IUT is made to request and display the contents of the Audit Log for target T1. The results are verified that they match the content of the log.

Test Configuration: The TD is setup with an AuditLog containing content from multiple sources and for multiple targets. The audit log contains example entries of all possible operations (see clause 19.Y.5) for audit target T1 with a mix of success and failure entries.

Test Steps:

1. WHILE (the IUT has not retrieved and displayed all entries for T1)

MAKE (the IUT request more content from the Audit Log)  
RECEIVE AuditLogQuery-Request  
    'Audit Log' = (the audit log in the TD),  
    'Query Parameters' = (a valid Audit Query by Target query including T1 as the target),  
    'Start At Sequence Number' = (any valid value)  
    'Requested Count' = (any valid value)  
TRANSMIT AuditLogQuery-Result  
    'Audit Log' = (the audit log in the TD),  
    'Records' = (the set of audit log records which match the query and which fit within the accepted response size),  
    'No More Items' = (TRUE if the last item is included, FALSE otherwise)

2. CHECK(that the displayed content matches audit records returned and that the complete records are presented)

Notes to Tester: If manual interaction is required between subsequent AuditLogQuery requests, checking of the displayed content might need to be performed before the manual interaction is taken instead of at the end of retrieving all of the items.