BACnet® TESTING LABORATORIES

INTERIM TEST SPECIFICATION

To Be Used with Test Package 18.1
Version 4
January 5, 2022

Approved by the BTL Working Group on December 7, 2022
Approved by the BTL Working Group Voting Members on January 14, 2022
Published on Januayr 17, 202

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

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.

2.1 Basic Functionality (Applies To All BACnet Devices)

JIRA: BTLWG-1144

The current Test Package does not test an IUT's ability to handle extended property identifiers as found in Protocol_Revision 20 and later.

For changes to the basic testing to be applied to all devices claiming Protocol_Revision 20 or later, see addendum Add-18.1-bk.

3.62 Staging Object

JIRA: BTLWG-421

The current Test Package does not include testing for the Staging object.

For BTL Checklist, BTL Test Plan, and BTL Specified Test changes for the testing the Staging object, and for Staging object changes to DS-COV-A, and DS-COV-B BIBBs, see Add-18.1-bd.

4.6 Data Sharing - WriteProperty - B

JIRA: BTLWG-1045

The current Test Package does not test that devices claiming Protocol_Revision 21 or later, accept writes of NULL to non-commandable properties.

Checklist Changes

None

Test Plan Changes

[In BTL Interim Tests, add into section 4.6.1, DS-WP-B Base Requirements]

BTL - 9.22.1.X3 - Writing NULL to Non-commandable Properties		
	Test Conditionality	If the IUT claims Protocol_Revision 20, or prior, this test shall be skipped. If the IUT does not contain any writable non-commandable properties, this test shall be skipped.
	Test Directives	Repeat the test for a selection of writable non-commandable, non-Present_Value properties which do not support the value NULL. Repeat the test for each object type with a writable non-commandable Present_Value supported by the IUT.
	Testing Hints	

[In DS-WP-B, Base Requirements, modify the entry for test 9.22.2.3]

BTL - 9.22.2.3 - Writing with a Property Value Having the Wrong Datatype		
	Test Conditionality	Must be executed.
	Test Directives	*Do not use NULL as the invalid datatype when applying the test.*
	Testing Hints	The pre-tester may want to consider running this test on a number of different properties with differing datatypes as this is a commonly failed test at the BTL.

4.46 Data Sharing - Lighting Output - B

JIRA: BTLWG-1168

The current Test Package does not include testing for this BIBB.

For Checklist, Test Plan, and BTL Specified Test changes for the testing of DS-LO-B, see addendum Add-18.1-misc.

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]

~~**BTL - 7.3.1.11 - Acked_Transitions Tests**~~ ***BTL - 7.3.1.11.1 - Acked_Transitions Test***		
	Test Conditionality	~~Must be executed.~~ *If the IUT only supports 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.*
	Test Directives	
	Testing Hints	

[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]

BTL - 7.3.1.11.2 - Acked_Transitions Test for Latching Objects		
	Test Conditionality	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.
	Test Directives	Apply the test for each supported transition.
	Testing Hints	

8.14 Device Management - Device Communication Control - B

JIRA: BTLWG-1145

The current Test Package does not properly test devices claiming a Protocol_Revision of 20 or greater.

For Checklist, Test Plan, and BTL Specified Test changes for devices claiming a Protocol_Revision of 20 or greater, see addendum Add_18.1_bi.

9.11 Data Link Layer - Virtual Network

JIRA: BTLWG-995

The current Test Package does not include testing of the Network Port requirements for virtual networks.

For Test Plan changes for devices claiming virtual network support, see addendum Add_18.1_misc.

13.5 Audit Reporting-View-A

JIRA: BTLWG-1132

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

These changes are not contained in any SSPC proposal.

Checklist Changes

[In BTL Checklist, replace AR-V-A section]

Audit Reporting - View - A		
	R‡	Base Requirements
	C[1]	Supports initiation of AuditLogQuery by Target
	C[1]	Supports initiation of AuditLogQuery by Source
	C[1]	Supports initiation of ReadRange
[1] 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

JIRA: BTLWG-1132

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

These changes are not contained in any SSPC proposal.

Checklist Changes

[In BTL Checklist, replace AR-AVM-A section]

Audit Reporting - Advanced View and Modify - A		
	R[‡]	Base Requirements
	C[1]	Supports initiation of AuditLogQuery by Target
	C[1]	Supports initiation of AuditLogQuery by Source
[1] At least one of these must be supported.		

Test Plan Changes

[In BTL Test Plan, replace section 13.6 Audit Reporting-Advanced View and Modify-A]

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.

Verify Checklist		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims AR-V-A
	Testing Hints	
Verify Checklist		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims DS-RP-A
	Testing Hints	
Verify Checklist		
	Test Conditionality	Must be executed.
	Test Directives	Verify that the IUT claims DS-WP-A
	Testing Hints	
Verify Checklist		
	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	
135.1-2019 - 8.18.3 - Reading and Presenting Properties		
	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	
135.1-2019 - 8.22.4 - Accepting Input and Modifying Properties		
	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

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

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.

9.22.1.X3 Writing NULL to Non-commandable Properties
JIRA: BTLWG-1045

Purpose: This test case verifies that the IUT returns a Result(+) when an attempt is made to relinquish a non-commandable property.

Test Concept: Write NULL to a writable non-commandable property, P1 in object O1, and verify the IUT returns a Result(+) and does not modify the property.

Test Configuration: P1 shall be a property for which NULL is not an accepted value.

Test Steps:

1. READ X = (O1), P1
2. TRANSMIT WriteProperty-Request,
'Object Identifier' = O1,
'Property Identifier' = P1,
'Property Value' = NULL
3. RECEIVE BACnet-SimpleACK-PDU
4. VERIFY (O1), P1 = X