



**BACnet® TESTING LABORATORIES
ADDENDA**

**Addendum br to
BTL Test Package 20.0.1**

**Revision v3
Revised 10/21/2022**

Approved by the BTL Working Group on 2022-07-28.
Approved by the BTL Working Group Voting Members on 2022-10-21.
Published on 2022-10-26.

[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-20.0.1 br-1: Add Intrinsic Fault Reporting to Lighting Output Objects [BTLWG-770].....2
BTL-20.0.1 br-2: Update Log_Interval Test per Interpretation Request [BTLWG-775]3
BTL-20.0.1 br-3: Accept writes of NULL to Non-Commandable Properties [BTLWG-1045]4

In the following document, language to be added to existing clauses within the BTL Test Package 20.0.1 is indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are proposed to be added, plain type is used throughout

In contrast, 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-20.0.1 br-1: Add Intrinsic Fault Reporting to Lighting Output Objects [BTLWG-770]

Overview:

135-2016br3 added intrinsic fault reporting to Lighting Output objects. This WI adds testing capability for it.

Changes:

Checklist Changes

[Modify the Lighting Output Object section to add new option]

Lighting Output Object	
O ³	Supports intrinsic reporting
¹ Protocol Revision 24 or higher must be claimed ² Contact BTL for interim tests for this functionality ³ Protocol Revision 21 or higher must be claimed	

Test Plan Changes

3.54 Lighting Output Object

[Add a new section in the test plan under Lighting Output Object]

3.54.13 Supports Intrinsic Reporting

The IUT supports intrinsic reporting.

Verify Checklist	
Test Conditionality	Must be executed.
Test Directives	Verify that the IUT claims support for AE-N-I-B in the Checklist with option "Implements the CHANGE_OF_RELIABILITY - NONE".
Testing Hints	

Specified Test Changes

None

BTL-20.0.1 br-2: Update Log_Interval Test per Interpretation Request [BTLWG-775]

Overview:

Jira item BTLWG-727. The expectations for timestamps are changed as per 135-2016br-9 and SSPC 135 official interpretation 135-2008-21.

Changes:

Checklist Changes

None

Test Plan Changes

None

Specified Test Changes

[Modify existing test 7.3.2.24.4 in BTL Specified Tests]

7.3.2.24.4 Log_Interval Test

Reason for Change: The Configuration Requirements are enhanced, and a Notes to Tester is added. **The expectations for timestamps are changed as per 135-2016br-9 and SSPC 135 official interpretation 135-2008-21.**

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

Purpose: To verify that the logging period is controlled by Log_Interval.

Test Concept: The logging object is configured to acquire data by polling. Polling is done at two different intervals, defined by Log_Interval, with about 10 records acquired at each rate. The timestamps of the records are inspected to verify the polling rate.

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. *Logging_Type is not equal to TRIGGERED.* Non-zero values shall be chosen for Log_Interval in accordance with the range and resolution specified by the manufacturer for this property.

Test Steps:

1. WRITE Log_Interval = (**L11**: some non-zero value)
2. WRITE Record_Count = 0
3. WAIT (**Internal Processing Fail Time** + 10* Log_Interval hundredths-seconds)
4. VERIFY (Log_Buffer record timestamp intervals **are L11, on average, are as written in step 1**)
5. WRITE Log_Interval = (**L12**: a non-zero value different from the one written in step 1)
6. WRITE Record_Count = 0
7. WAIT (**Internal Processing Fail Time** + 10* Log_Interval hundredths-seconds)
8. VERIFY (Log_Buffer record timestamp intervals **are L12, on average, are as written in step 5**)

Notes to Tester: The step 1 write of Logging_Interval to a non-zero value will make a change in Logging_Type from COV to POLLED, if Logging_Type was initially COV.

BTL-20.0.1 br-3: Accept writes of NULL to Non-Commandable Properties [BTLWG-1045]

Overview:

Interim testing is needed for 135-2016br-2 - accept writes of NULL to non-commandable properties

Changes:

Checklist Changes

None

Test Plan Changes

[In Test Plan, add into section 4.6.1, DS-WP-B Base Requirements, the yellow highlight was an addition made per voters comments.]

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 which do not support the value NULL.
	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.

Specified Test Changes

[Add new test to BTL Specified Tests]

9.22.1.X3 Writing NULL to Non-commandable Properties

Reason for Change: The standard was changed in PR21 to require that devices not return errors when a NULL is written to non-commandable properties and no test exists for this functionality.

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