



**BACnet[®] TESTING LABORATORIES
ADDENDA**

**Addendum Errata to
BTL Test Package 23.0**

**Revision v2
Revised 4/26/2023**

Approved by the BTL Working Group on March 30, 2023;
Approved by the BTL Working Group Voting Members on March 30, 2023;
Published on April 28, 2023.

[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-23.0 Errata-1: Correct typo in BTL-9.20.2.1 [BTLWG-1328].....2

BTL-23.0 Errata-2: Correct Tests 7.3.2.24.X8 and 7.3.2.24.X9 [BTLWG-1378].....3

BTL-23.0 Errata-3: Correct Test 135.1-7.3.1.15 [BTLWG-1379]5

BTL-23.0 Errata-4: Correct Test 9.X35.1 time reference [BTLWG-1385]7

BTL-23.0 Errata-5: Correct use of Test 14.1.X11 in Test Plan [BTLWG-1390]9

BTL-23.0 Errata-6: Correct Test Reference for Verify Tpostdrive in Test Plan [BTLWG-1400] 10

BTL-23.0 Errata-7: Update Checklist to include PR25 Items [BTLWG-1412] 11

In the following document, language to be added to existing clauses within the BTL Test Package 23.0 is indicated through the use of *italics*, while deletions are indicated by ~~strike through~~. Where entirely new subclauses are proposed to be added, plain type is used throughout

In 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-23.0 Errata-1: Correct typo in BTL-9.20.2.1 [BTLWG-1328]

Overview:

From test case 9.20.2.1 Reading a Single, Unsupported Property from a Single Object, the "OR" symbol is missing from step 2.

Note: This is a 135.1 Errata

Changes:

Checklist Changes

None

Test Plan Changes

In Section 4.4.1 Base Requirements for Data Sharing - ReadPropertyMultiple - B

Change reference from 135.1 to BTL

<i>BTL135.1 2019 - 9.20.2.1 - Reading a Single, Unsupported Property from a Single Object</i>		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

Specified Test Changes

[Move 135.1 test 9.20.2.1 to BTL to make correction]

9.20.2.1 Reading a Single, Unsupported Property from a Single Object

Reason for Change: Errata type to add missing '|' (or) to step 2.

Purpose: To verify the ability to correctly execute a ReadPropertyMultiple service request for which the 'List of Read Access Specifications' contains specifications for a single unsupported property.

Test Steps:

1. TRANSMIT ReadPropertyMultiple-Request,
 'Object Identifier' = Object1 | Object2,
 'Property Identifier' = (any property, P1, that is not supported in the selected object)
2. RECEIVE BACnet-Error-PDU,
 'Error Class' = PROPERTY,
 'Error Code' = UNKNOWN_PROPERTY
 (ReadPropertyMultiple-ACK,
 'Object Identifier' = (the object identifier from step 1),
 'Property Identifier' = P1,
 'Error Class' = PROPERTY,
 'Error Code' = UNKNOWN_PROPERTY)

BTL-23.0 Errata-2: Correct Tests 7.3.2.24.X8 and 7.3.2.24.X9 [BTLWG-1378]

Overview:

Tests 7.3.2.24.X8 and 7.3.2.24.X9 in BTL Specified Tests 23.0 refer to the property "Logging_Mode" should be "Logging_Type".

Changes:

Checklist Changes

None

Test Plan Changes

None

Specified Test Changes

[Modify Test 7.3.2.24.X8 and 7.3.2.24.X9 to correct property reference]

7.3.2.24.X8 Clock-Aligned Logging

Reason for Change: No test available for this functionality. This test is now in Addenda s and needs the errata correction show below.

Purpose: To verify that logged trend records have timestamps aligned to that interval, when Align_Intervals is TRUE and Log_Interval is a factor of (divides without remainder) a day.

Test Concept: For this test, select two evenly divisible factors. Write each to Log_Interval in the test. Trend records are logged, and checked that those are aligned to the Log_Interval. This is done twice to ensure that different interval frequency behavior is verified. This test does not employ Log_Interval values which are not one of the evenly divisible factors.

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 in order that it occurs after the end of the test. Stop_When_Full, if configurable, shall be set to FALSE. Enable is initially FALSE. Interval_Offset is set to zero. Align_Intervals is set to TRUE. Triggering of non periodic log records should not occur during this test. ~~Logging_Mode~~Logging_Type is POLLED. X1 and X2 are each a value which the IUT supports for which the standard mandates the alignment behavior.

Notes to Tester: The values for Log_Interval which require alignment are those for which the standard mandates the alignment behavior, where 8,640,000 modulo Log_Interval is zero.

Test Steps:

1. CHECK (Log_Buffer contains 1 entry, and it is the buffer-purged event)
2. WRITE Log_Interval = X1
3. WRITE Enable = TRUE
4. MAKE (logging object collect at least 2 records)
5. WRITE Enable = FALSE
6. CHECK (Log_Buffer contains at least 5 entries, and at least two data records)
7. CHECK (that the timestamp of each data record, since the Log_Interval was written, is a multiple of X1)
8. WRITE Log_Interval = (X2, any value which requires alignment behavior, that was not already chosen)
9. WRITE Enable = TRUE
10. MAKE (logging object collect at least 2 more records)
11. WRITE Enable = FALSE
12. CHECK (Log_Buffer has collected two or more additional data records and two or more log-status records)
13. CHECK (that the timestamp are multiples of X2 for all data records collected, since the write with X2)

7.3.2.24.X9 Logging Interval_Offset

Reason for Change: No test for this functionality. This test is now in Addenda s and needs the errata correction show below.

Purpose: To verify that timestamps abide by the Interval_Offset.

Test Concept: Log_Interval is set to a value which the IUT supports which is a factor of (divides without remainder) a day and which is greater than 3 seconds.

Interval_Offset is first set to a non-zero value less than Log_Interval. After logging some records, their timestamps are checked. The logging is stopped. Interval_Offset is set to a value which the IUT supports greater than Log_Interval, logging is re-enabled, and the timestamps again are checked.

Configuration Requirements: Align_Intervals is set to TRUE. The Log_DeviceObjectProperty property in a Trend Log or in a Trend Log Multiple, is configured to the property or properties monitored. 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 in order that it occur after the end of the test. Stop_When_Full, if configurable, shall be set to FALSE. Enable is initially FALSE. Align_Intervals is set to TRUE. Triggering of non periodic log records should not occur during this test. If the Interval_Offset cannot be set to a value which the IUT supports greater than Log_Interval, then steps 11 through the end of this test are skipped.

~~Logging_Mode~~Logging_Type is POLLED. An evenly divisible value is a value for which the standard mandates the alignment behavior.

Notes to tester: Interval_Offset in logging objects, and Log_Interval are each an Unsigned number of hundredths of seconds. Excellent choices are 400, 500, 600, 1000, or 1200. When Interval_Offset is larger than Log_Interval, then Interval_Offset modulo Log_Interval, is smaller than Log_Interval.

Test Steps:

1. WRITE Record_Count = 0
2. CHECK (Log_Buffer contains 1 entry, and it is the buffer-purged event)
3. WRITE Log_Interval = (any evenly divisible value greater than 3 seconds)
4. WRITE Interval_Offset = (any value, between 2 seconds and Log_Interval - 1 seconds)
5. WRITE Enable = TRUE
6. MAKE (logging object collect at least 2 records)
7. WRITE Enable = FALSE
8. CHECK (Log_Buffer contains two or more data records and at least three log-status)
9. CHECK (the timestamp for the data records have a fixed offset, determined by Log_Interval and Interval_Offset)
10. WRITE Interval_Offset = (any value greater than Log_Interval)
11. WRITE Enable = TRUE
12. MAKE (logging object collect at least 2 records)
13. WRITE Enable = FALSE
14. CHECK (Log_Buffer has collected two or more additional data records and two or more log-status entries)
15. CHECK (the timestamp for data records collected since the Interval_Offset was last written, have Log_Interval between records, at a fixed offset of Interval_Offset modulo Log_Interval)

BTL-23.0 Errata-3: Correct Test 135.1-7.3.1.15 [BTLWG-1379]**Overview:**

135.1 7.3.1.15 Number_Of_States Range Test from 135_1_2019_final_20190828.pdf is inconsistent in how it names its property:

Step 2: Uses "Number_Of_States" instead of "COUNT"

(it read Number_Of_States into COUNT variable in step 1). Step 3/4/8 correctly refer to "COUNT".

Step 5 and 8: P1 should be Present_Value

Changes:

Checklist Changes

None

Test Plan Changes

[Modify section 3.14 Multi-state Input Object in Base Requirements]

BTL135.1-2019 - 7.3.1.15 Number Of States Range Test		
	Test Conditionality	If the IUT does not contain an object with a writable, or conditionally writable Present_Value, this test shall be skipped.
	Test Directives	
	Testing Hints	

Specified Test Changes

[Move test 7.3.1.15 from 135.1 to BTL Specified Tests to correct property usage.]

7.3.1.15 Number_Of_States Range Test

Reason for Change: correct property usage.

Dependencies: ReadProperty Service Execution Tests, 7.1; WriteProperty Service Execution Tests 7.2.2

BACnet Reference Clauses: 12.18.11, 12.19.4, 12.19.11, 12.20.4, 12.20.10

Purpose: This test case verifies that the Present_Value property of a Multi-state Input, Multi-state Output or Multi-state Value object is limited to the range 1 through the value of its Number_Of_States property.

Test Concept: The Number_Of_States property is first verified to be a non-zero value. An attempt is then made to modify the Present_Value property of the referenced object to contain a value greater than the value of the Number_Of_States property. An attempt is then made to set the Present_Value property to 0. The test shall verify a correct error response for each case.

Configuration Requirements: The IUT shall be configured to contain a Multi-state Input, Multi-state Output, or Multi-state Value object, Object1, which contains a writable Present_Value property. Object1 shall be configured such that the Present_Value property is writable before executing the test. If the Present_Value property cannot be made writable, then this test shall be skipped.

Test Steps:

1. READ COUNT = Number_Of_States
2. VERIFY **COUNT** ~~Number_Of_States~~ = (a non-zero value)

3. TRANSMIT WriteProperty-Request,
 'Object Identifier' = Object1,
 'Property Identifier' = Present_Value,
 'Property Value' = (any value larger than COUNT)
4. RECEIVE BACnet-Error-PDU,
 Error Class = PROPERTY,
 Error Code = VALUE_OUT_OF_RANGE
5. VERIFY (Object1), *Present_Value* = (a value between 1 and COUNT, inclusive)
6. TRANSMIT WriteProperty-Request,
 'Object Identifier' = Object1,
 'Property Identifier' = Present_Value,
 'Property Value' = 0
7. RECEIVE BACnet-Error-PDU,
 Error Class = PROPERTY,
 Error Code = VALUE_OUT_OF_RANGE
8. VERIFY (Object1), *Present_Value* = (a value between 1 and COUNT, inclusive)

Note To Tester: When testing an object that is commandable, any priority may be selected.

BTL-23.0 Errata-4: Correct Test 9.X35.1 time reference [BTLWG-1385]

Overview:

In Test 9.X35.1 Uses Who-Is to Configure Devices Supporting the Who-Am-I Service in step 4 uses "Unconfirmed Request Fail Time". But there is no definition for that time. Change reference to "Internal Processing Fail Time".

Changes:

Checklist Changes

None

Test Plan Changes

None

Specified Test Changes

9.X35.1 Uses Who-Is to Configure Devices Supporting the Who-Am-I Service

Reason for Change: No test exists for this functionality.

Purpose: To verify that the IUT can configure a device when it receives a Who-Am-I in response to a Who-Is that it sent.

Test Concept: The IUT sends a Who-Is and TD responds with a Who-Am-I. The IUT then configures the TD's Device object instance number (X) using the You-Are service.

Configuration Requirements: TD has a MAC address but is not configured with a Device object instance number.

Notes to Tester: The IUT may require the tester to specify a Device object instance number to assign TD, using the IUT's software. If the IUT requires this step to occur at the beginning of step 3, then the amount of time it takes the tester to enter this should not be counted towards the fail timer. The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. MAKE (the IUT transmit a Who-Is request to discover devices needing configuration)
2. RECEIVE
DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
Who-Is Request,
'Device Instance Range Low Limit' = 4194303,
'Device Instance Range High Limit' = 4194303
3. TRANSMIT
DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
Who-Am-I Request,
'Vendor Identifier' = (TD's Vendor_Identifier),
'Model Name' = (any valid Model Name),
'Serial Number' = (any valid Serial Number)
4. BEFORE **Internal Processing** ~~Unconfirmed Request~~ Fail Time
RECEIVE
DESTINATION = TD | GLOBAL BROADCAST | LOCAL BROADCAST | REMOTE BROADCAST,
You-Are Request,
'Vendor Identifier' = (TD's Vendor_Identifier),
'Model Name' = (the Model Name sent by TD in the previous step),

'Serial Number' = (the Serial Number sent by TD in the previous step),

'Device Identifier' = (Device, X),

'Device MAC Address' = (TD's MAC address, or absent)

5. IF (TD is not an MS/TP subordinate node)

TRANSMIT

DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,

I-Am Request,

'Device Identifier' = (Device, X),

'Max APDU Length Accepted' = (any valid value),

'Segmentation Supported' = (any valid value),

'Vendor Identifier' = (TD's Vendor_Identifier)

BTL-23.0 Errata-5: Correct use of Test 14.1.X11 in Test Plan [BTLWG-1390]

Overview:

Test 14.1.X11 “Processing Forwarded-NPDU request initiated from different port” is referenced in Test Plan 9.3.2 and 9.3.4. This test is a non-BBMD test and should not be referenced 9.3.4.

Changes:

Checklist Changes

None

Test Plan Changes

[In the Test Plan, section 9.3.4 Is Able to Operate in BBMD Mode should remove the following test entry from the list of tests in this section.]

BTL – 14.1.X11 – Processing Forwarded NPDU request initiated from different port		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

Specified Test Changes

[None]

BTL-23.0 Errata-6: Correct Test Reference for Verify Tpostdrive in Test Plan [BTLWG-1400]

Overview:

In BTL Test Plan 23.0 chapter 9.2.1 there is a test reference:

135.1-2019 - 12.3.3.2 - Verify Tpostdrive

but in 135.1-2019 this number does not exist.

There is a test 12.1.3.2 Verify Tpostdrive though.

Changes:

BTL Checklist Changes

[None]

BTL Test Plan Changes

[In BTL Test Plan, Correct test reference]

9.2 Data Link Layer - MS/TP - Slave Node

9.2.1 Base Requirements

Base Requirements for all MS/TP slave devices.

135.1-2019 - 12.1.3.2 12.3.3.2 - Verify T _{postdrive}		
	Test Conditionality	Must be executed.
	Test Directives	Every MS/TP device shall claim 9600 baud. Test that the device operates at each baud rate that is claimed. Devices claiming Protocol_Revision 12 or higher where Addendum 135-2008ab is incorporated, shall claim 38400.
	Testing Hints	

Test Changes

None

BTL-23.0 Errata-7: Update Checklist to include PR25 Items [BTLWG-1412]**Overview:**

The Checklist needs to be updated to include changes from PR 25.

BTL Checklist Changes

Data Link Layer - Secure Connect		
	R	Base Requirements
	C ¹	Is able to operate as a node without a local hub function
	C ¹	Is able to operate as a hub
	O	Supports direct connections
	O ²	Is able to accept direct connections
	O ²	Is able to initiate direct connections
	O	Supports configuration through Network Port object
	C ^{3,4}	<i>Supports Data Attributes as of Protocol Revision 25</i>
¹ At least one of these options must be supported. ² At least one of these options must be supported if the device supports direct connections. ³ <i>Required if the IUT claims Protocol Revision 25 or higher.</i> ⁴ <i>Contact BTL for interim tests for this functionality.</i>		

BTL Test Plan Changes

9.9.8 Supports Data Attributes as of Protocol Revision 25

Contact BTL for interim tests for this functionality.

Test Changes

None