



BACnet[®] TESTING LABORATORIES ADDENDA

Addendum crs1 to BTL Test Package 20.0

**Revision v2
Revised 4/7/2022**

Approved by the BTL Working Group on April 7, 2022;
Approved by the BTL Working Group Voting Members on May 17, 2022;
Published on June 16, 2022..

[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 crs1-1: Update Language Regarding Non-existent Objects in Test 9.18.2.3 [BTLWG-960, CR-0245].....	2
BTL-20.0 crs1-2: Total_Record_Count Testing [BTLWG-965, CR-0274]	4
BTL-20.0 crs1-3: Change Test Conditionality per CR-0508 [BTLWG-1170, CR-0508]	6
BTL-20.0 crs1-4: Accepts Segmented Response Section Description Changes [BTLWG-1175].....	7
BTL-20.0 crs1-6: IPv6 Distribute-Broadcast-To-Networks [BTLWG-1226, CR-0521].....	8
BTL-20.0 crs1-7: Access Point Object Test 7.3.2.X56.1 Conditionality Change [BTLWG-1141, CR-0502]	11
BTL-20.0 crs1-8: Corrections for BTL Test 12.X.4.2.1 Reject Forwarded NPDU [BTLWG-1200]	12

In the following document, language to be added to existing clauses within the BTL Test Package 20.0 is indicated through the use of *italics*, while deletions are indicated by ~~striketrough~~. 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 crs1-1: Update Language Regarding Non-existent Objects in Test 9.18.2.3 [BTLWG-960, CR-0245]**Overview:**

In response to CR-0245, the BTL-WG agreed to update test 9.18.2.3 to allow return of UNSUPPORTED_OBJECT_TYPE in addition to UNKNOWN_OBJECT. Reference to test 9.20.1.6 is given as it did allow this.

Subsequently, 135.1 has been changed address this issue in 9.20.1.6 in a different manner by being more specific that supported object types should be checked in the test, not unsupported ones.

Changes:

Checklist Changes

None

Test Plan Changes

[Update reference for test 9.18.2.3 in the BTL Test Plan document]

4.2 Data Sharing - ReadProperty - B**4.2.1 Base Requirements**

All devices must support this BIBB.

135.1-2019 BTL - 9.18.2.3 - Reading an Unknown Object		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

Specified Test Changes

[Move test 9.18.2.3 from 135.1-2019 to BTL and make the changes as noted below.]

9.18.2.3 Reading an Unknown Object

Reason for Change: More specific test for a non-existing object that is of a type supported by the IUT or an object that is of a type not supported by the IUT.

Purpose: To verify that the IUT can execute ReadProperty service requests under circumstances where the requested object does not exist.

Test Concept: ~~The TD attempts to read a property of an object that does not exist in the device.~~ *The TD first attempts to read from a non-existent object of a type supported by the IUT. The returned error class/code is verified to be OBJECT/UNKNOWN OBJECT. The TD then attempts to read from a non-existent object of a type which is not supported by the IUT. The returned error class/code is verified to be OBJECT/UNKNOWN_OBJECT or OBJECT/UNSUPPORTED_OBJECT_TYPE.*

Test Steps:

- TRANSMIT ReadProperty-Request,
'Object Identifier' = *(any non-existent object, which is of a type supported by the IUT)* ~~any standard object not contained in the IUT's database~~,
'Property Identifier' = (any property defined for the specified object)
- RECEIVE BACnet-Error-PDU,
Error Class = OBJECT,
Error Code = UNKNOWN_OBJECT
- TRANSMIT ReadProperty-Request.*

'Object Identifier' = (any object of a type not supported by the IUT),

'Property Identifier' = (any property defined for the specified object)

4. *RECEIVE BACnet-Error-PDU_s*

Error Class = OBJECT_s

Error Code = UNSUPPORTED_OBJECT_TYPE | UNKNOWN_OBJECT

BTL-20.0 *crs1-2*: Total_Record_Count Testing [BTLWG-965, CR-0274]

Overview:

CR-0274 noted that no test ensures that Total_Record_Count is not reset when Record-Count is written to 0.

Changes:

Checklist Changes

None

Test Plan Changes

[Change the 5 references to 7.3.2.24.9 Total_Record_Count Test from 135.1-2019 to BTL]

Specified Test Changes

[Move test from 135.1-2019 to BTL Specified Test and apply changes noted below]

7.3.2.24.9 Total_Record_Count Test

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

BACnet Reference Clause: 12.25.16.

Purpose: To verify that the Total_Record_Count property increments for each record added to the Log_Buffer, even after Buffer_Size records have been added. (Note: it is not reasonable to test for the requirement of BACnet Clause 12.25.16 that the value wrap from $2^{32}-1$ to one; even if a record was collected every 100th of a second it could take more than 497 days to complete the test.)

Test Concept: The logging object is configured to acquire data by whatever means. *Total_Record_Count* is read to determine an initial value. Record_Count is set to zero and Total_Record_Count is read. *It is verified that Total_Record_Count is incremented and not reset to 0 when Record_Count is written to 0.* Collection of data proceeds until Record_Count changes, collection is halted and Total_Record_Count is checked that it has incremented by Record_Count. If, for whatever reason, the IUT cannot be configured such that the TD is able to halt collection before Buffer_Size records are collected this test shall not be performed.

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. Enable shall be set to FALSE.

Test Steps:

1. READ TRC1 = Total_Record_Count
2. ~~1.~~ WRITE Record_Count = 0
3. ~~2.~~ WAIT Internal Processing Fail Time
4. VERIFY Total_Record_Count = TRC1 + 1
5. ~~3.~~ READ X = Total_Record_Count
6. ~~4.~~ READ Y = Record_Count
7. ~~5.~~ WRITE Enable = TRUE
8. ~~6.~~ WHILE (Record_Count = Y + 1) DO { }
9. ~~7.~~ WRITE Enable = FALSE
10. ~~8.~~ WAIT Internal Processing Fail Time
11. ~~9.~~ IF (Total_Record_Count - X != Record_Count - Y) THEN
 ERROR "Total_Record_Count has incorrect value."

BTL-20.0 crs1-3: Change Test Conditionality per CR-0508 [BTLWG-1170, CR-0508]**Overview:**

In the BACnet section 12.13.6 the BACnet Standard does not require arbitrary truncation or expansion by writing property File_Size even if the File_Size can be changed by writing zero or by replacement of the entire file.

The Test 9.13.1.2.4 request in Step 5 to Write File_Size a value > 0

Changes:

Checklist Changes

None

Test Plan Changes

[Change Test Conditionality for test 9.13.1.2.4 as shown below]

3.61 File Object**3.61.5 Contains a Writable Stream-Based File for a Purpose Other Than Backup and Restore**

135.1-2019 - 9.13.1.2.4 - Truncating a File		
	Test Conditionality	If the file size cannot be changed, then this test shall be skipped. <i>If the only value that the IUT accepts when writing to File_Size is zero, then this test shall be skipped</i>
	Test Directives	
	Testing Hints	

Specified Test Changes

None

BTL-20.0 crs1-4: Accepts Segmented Response Section Description Changes [BTLWG-1175]**Overview:**

In the optional section 2.2.3 Accepts Segmented Responses without Specifying the Maximum, the description:

BACnet devices which accept segmented responses shall meet these requirements.

inappropriately implies that all devices need to meet this optional section. The description should be changed to:

The IUT supports receiving segmented responses and is able to generate requests with max-segments-accepted set to B'000'.

Changes:

Checklist Changes

None

Test Plan Changes

[Modify the description of the section 2.2.3 Accepts Segmented Responses without Specifying the Maximum to the suggested wording]

2.2.3 Accepts Segmented Responses without Specifying the Maximum

~~BACnet devices which accept segmented responses shall meet these requirements.~~ *The IUT supports receiving segmented responses and is able to generate requests with max-segments-accepted set to B'000'.*

BTL - 13.1.12.X1 - Reading with maximum-segments-accepted bit pattern B'000'		
	Test Conditionality	If the IUT cannot be configured to issue any BACnet-Confirmed-Request-PDU with 'segmented-response-accepted' = TRUE and the 'max-segments-accepted' parameter equal to B'000', then this test shall be skipped.
	Test Directives	
	Testing Hints	

Specified Test Changes

None

BTL-20.0 *crs1-6*: IPv6 Distribute-Broadcast-To-Networks [BTLWG-1226, CR-0521]

Overview:

CR-0521 pointed out 3 errors in test 12.X.4.1.5 Distribute-Broadcast-To-Network:

- 1) In all of the Forwarded-NPDU messages, the field Source-Virtual-Address is instead of the field Original-Source-B/IPv6-Address.
- 2) In step 6, the IUT is expected to forward the WhoIs to FD2, but FD2 is not registered with the IUT.
- 3) In step 7, the IUT is expected to generate a local broadcast I-Am response, but the standard allows unicast and global broadcast I-Am responses.

For issue 2, this proposal changes the test to have FD2 register with the IUT.

For issue 3, the proposal adds in other allowable options for I-Am generation, but also removes all of the associated forwarding of the I-Am to the other BBMDs and FDs. This approach is taken as the important point is that the Who-Is was pushed to the IUT's local application layer and the generation of the I-Am shows this. The subsequent distribution of the IAm to other BBMDs and foreign devices is not important to the test.

Changes:

Checklist Changes

None

Test Plan Changes

None

Specified Test Changes

Note: the strikethrough in the following test should be removed when the content is placed into the BTL Specified Tests. It is included here solely to highlight the deletions that were made to the BTL version of the test

[Modify existing test with the following changes.]

12.X.4.1.5 Distribute-Broadcast-To-Network

Reason for Change: New to standard.

Purpose: To verify that the IUT, configured as a BBMD, will process a Distribute-Broadcast-To-Network request.

Test Concept: Send a Distribute-Broadcast-To-Network message containing a Who-Is request to the IUT from a registered foreign device. Verify that the IUT distributes it to all associated BBMDs and registered foreign devices. Also verify that the IUT processes the Who-Is request by checking that the IUT responds with an I-Am.

Configuration Requirements: Register FD1 and FD2 as a foreign devices with the IUT. ~~FD2 is a registered foreign device with BBMD1. For purposes of this test, TD is acting as FD1.~~

~~Steps 1-6 are the processing of the Distributed-Broadcast-To-Network, Step 7 and on is the processing of the APDU service by the IUT.~~

Notes to Tester: The order of the forwarded messages transmitted by the IUT is not significant.

Test Steps:

1. TRANSMIT

DA = IUT,
SA = FD1,
Distribute-Broadcast-To-Network,
Who-Is-Request

-- verify the broadcast is sent to the local IPv6 multicast address

2. RECEIVE

DA = B/IPv6 Link Local Multicast Address,
SA = IUT,
Forwarded-NPDU,
~~Source Virtual Address = FD1,~~
Original-Source-Virtual-Address = FD1,
Original-Source-B/IPv6-Address = FD1,
Who-Is-Request

-- verify the broadcast is sent to the broadcast to each peer BBMD

3. RECEIVE

DA = BBMD1,
SA = IUT,
Forwarded-NPDU,
~~Source Virtual Address = FD1,~~
Original-Source-Virtual-Address = FD1,
Original-Source-B/IPv6-Address = FD1,
Who-Is-Request

4. RECEIVE

DA = BBMD2,
SA = IUT,
Forwarded-NPDU,
~~Source Virtual Address = FD1,~~
Original-Source-Virtual-Address = FD1,
Original-Source-B/IPv6-Address = FD1,
Who-Is-Request

5. RECEIVE

DA = BBMD3,
SA = IUT,
Forwarded-NPDU,
~~Source Virtual Address = FD1,~~
Original-Source-Virtual-Address = FD1,
Original-Source-B/IPv6-Address = FD1,
Who-Is-Request

-- verify the broadcast is sent to all other registered foreign devices

6. RECEIVE

DA = FD2,
SA = IUT,
Forwarded-NPDU,
~~Source Virtual Address = FD1,~~
Original-Source-Virtual-Address = FD1,
Original-Source-B/IPv6-Address = FD1,
Who-Is-Request

7. CHECK(that the IUT does not send the Who-Is request to FD1)

-- verify that the IUT sent the Who-Is to its own application layer as well by verifying

-- it responds to the request with an I-Am

8. RECEIVE

DA = B/IPv6 Link Local Multicast Address,
SA = IUT,

Original-Broadcast-NPDU,
 Original-Source-Virtual-Address = IUT,
 DNET = 65535 or absent,
 I-Am-Request

| (
 DA = FD1,
 SA = IUT,
 Original-Unicast-NPDU,
 Source-Virtual-Address = IUT,
 Destination-Virtual-Address = FD1,
 I-Am-Request
)

8. RECEIVE

DA = BBMD1,
 SA = IUT,
 Forwarded-NPDU,
 Source-Virtual-Address = IUT,
 Original-Source-Virtual-Address = IUT,
 I-Am-Request

9. RECEIVE DA = BBMD2,

SA = IUT,
 Forwarded-NPDU,
 Source-Virtual-Address = IUT,
 Original-Source-Virtual-Address = IUT,
 I-Am-Request

10. RECEIVE DA = BBMD3,

SA = IUT,
 Forwarded-NPDU,
 Source-Virtual-Address = IUT,
 Original-Source-Virtual-Address = IUT,
 I-Am-Request

11. RECEIVE

DA = FD1,
 SA = IUT,
 Forwarded-NPDU,
 Source-Virtual-Address = IUT,
 Original-Source-Virtual-Address = IUT,
 I-Am-Request

12. RECEIVE

DA = FD2,
 SA = IUT,
 Forwarded-NPDU,
 Source-Virtual-Address = IUT,
 Original-Source-Virtual-Address = IUT,
 I-Am-Request

BTL-20.0 crs1-7: Access Point Object Test 7.3.2.X56.1 Conditionality Change [BTLWG-1141, CR-0502]

Overview:

Test assumes a writable Out_Of_Service property to run correctly. We need a conditionality to skip the test if this is not possible.

Changes:

Checklist Changes

None

Test Plan Changes

[Modify the Test Conditionality for test 7.3.2.X56.1 as shown below]

3.44.1 Base Requirements

Base requirements must be met by any IUT that can contain Access Point objects

BTL - 7.3.2.X56.1 - Authentication Status and Access Event Test		
	Test Conditionality	Must be executed. <i>This test shall be skipped if Out_Of_Service is not writable and cannot otherwise be made TRUE</i>
	Test Directives	
	Testing Hints	

Specified Test Changes

None

BTL-20.0 crs1-8: Corrections for BTL Test 12.X.4.2.1 Reject Forwarded NPDU [BTLWG-1200]**Overview:**

Test 12.X.4.2.1 Reject Forwarded-NPDU should not use FD3 as the source of the forwarded message as FD3 is registered with the IUT. A different foreign device, which is registered with BBMD1, should be the source of the message.

Changes:

Checklist Changes

None

Test Plan Changes

[Change the test name for 12.X.4.2.1 as shown below]

9.8.4 Is Able to Operate in BBMD Mode

...

...		
BTL - 12.X.4.1.5 - Distribute-Broadcast-To-Network		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
BTL - 12.X.4.2.1 - Reject Forwarded-NPDU		
BTL - 12.X.4.2.1 - Ignore Forwarded-NPDU from non-Participating BBMDs		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
BTL - 12.X.4.2.2 - Reject Address-Resolution		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
...		

Specified Test Changes

[Modify the existing test by changing the name and other changes as shown in yellow highlights. Note there is no italic because this test is a BTL specified test only. The strikethrough is for reference but will not be put in the BTL Specified Test document.]

12.X.4.2.1 ~~Reject Forwarded-NPDU~~**12.X.4.2.1 Ignore Forwarded-NPDU from non-Participating BBMDs**

Reason for Change: Fixed the test to reference a different foreign device.

Purpose: To verify that the IUT, configured as a BBMD, will ~~ignore~~~~drop~~ a Forwarded-NPDU request from a BBMD that's not in the IUT's BDT.

Test Concept: The IUT is configured as a BBMD and is actively forwarding messages to registered devices. Validate that the IUT does not forward a message received from a BBMD not listed in the IUT's BDT.

Configuration Requirements: TD shall operate as BBMD4 and is not listed in ~~Empty~~ the IUT's BDT. **FD13** is a foreign device registered with **BBMD4**~~the IUT~~. The IUT is configured with at least one foreign device.

Test Steps:

1. TRANSMIT
DA = IUT,
SA = BBMD41,
Forwarded-NPDU,
Source-Virtual-Address = FD13,
Original-Source-B/IPv6-Address = FD13
I-Am-Request
2. CHECK (The IUT does not forward the message to any foreign devicesissue any Forwarded NPDU BVLCs)