



**BACnet® TESTING LABORATORIES  
ADDENDA**

**Addendum cc to  
BTL Test Package 23.3**

**Revision final  
Revised 10/3/2024**

Approved by the BTL Working Group on August 22, 2024;  
Approved by the BTL Working Group Voting Members on October 2, 2024;  
Published on October 7, 2024.

**[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-23.3 cc-1: Update Network Port Object and Add BACnet/SC Configuration Support [BTLWG-1372, 1492, and Interim Checklist Changes] .....	2
BTL-23.3 cc-2: BACnet/SC Heartbeat and Hub Function Tests [BTLWG-1429, BTLWG-1469].....	23
BTL-23.3 cc-3: DISCARD_CHANGES Command Testing [BTLWG-1493].....	29
BTL-23.3 cc-4: Fix Direct Connect Establishment Test [BTLWG-1517].....	31
BTL-23.3 cc-5: New Tests for Network Port Object and B/SC [BTLWG-1477].....	33
BTL-23.3 cc-6: DISCARD_CHANGES Command with Non-Writable NPOs [BTLWG-1571] .....	38

In the following document, language to be added to existing clauses within the BTL Test Package 23.3 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-23.3 cc-1: Update Network Port Object and Add BACnet/SC Configuration Support [BTLWG-1372, 1492, and Interim Checklist Changes]**

**Overview:**

Per 135-2020cc-1. Update the Network Port Object and Add BACnet/SC Configuration Support

Per 1492, Tests 7.3.2.46.3.X.1, 19.Y.3.1, 19.Y.3.2, and 19.Y.3.3 do not test the File object Modification\_Date and File\_Size properties.

**Changes:**

---

**Checklist Changes**

---

[Modify the Objects section]

File Object		
	R	Base Requirements
	C <sup>1</sup>	Supports DM-BR-B
	C <sup>1,2</sup>	Supports BACnet/SC Certificate Exchange
	C <sup>1</sup>	Supports a record-based File object for a purpose other than Backup and Restore <i>or</i> BACnet/SC Certificate Exchange
	C <sup>1</sup>	Supports a stream-based File object for a purpose other than Backup and Restore <i>or</i> BACnet/SC Certificate Exchange
	O	Contains a writable stream-based File for a purpose other than Backup and Restore <i>or</i> BACnet/SC Certificate Exchange
<sup>1</sup> At least one of these options is required if the IUT supports the File object type. <sup>2</sup> Protocol Revision 24 or higher must be claimed		

Network Port Object		
	R <sup>1</sup>	Base Requirements
	C <sup>2</sup>	Supports writable Network_Number property
	S	Supports configurable Out_Of_Service property
	C <sup>3</sup>	Supports Non-hierarchical Network Port objects
	C <sup>3</sup>	Supports hierarchical Network Port objects
	O	Supports the Command property
	O <sup>3,4</sup>	Supports the DISCARD_CHANGES command
	O <sup>3,4</sup>	Supports the RENEW_FD_REGISTRATION command
	O <sup>3,4</sup>	Supports the RESTART_SLAVE_DISCOVERY command
	O <sup>3,4</sup>	Supports the RENEW_DHCP command
	O <sup>3,4</sup>	Supports the RESTART_AUTONEGOTIATION command
	O <sup>3,4</sup>	Supports the DISCONNECT command
	O <sup>3</sup>	Supports the RESTART_PORT command
	O <sup>4,5</sup>	Supports the GENERATE_CSR_FILE command
	O <sup>4,5</sup>	Supports the VALIDATE_CHANGES command
	O	Supports the Routing_Table property
<sup>1</sup> Support for Network Port objects is required for devices claiming Protocol_Revision 17 or higher. <sup>2</sup> Support for writable Network_Number properties is required in routers and other devices that need to know the network number in order to operate. <sup>3</sup> At least one of these options is required <sup>3,4</sup> At least one of these options is required if the Command property is supported. <sup>5</sup> Protocol Revision 24 or higher must be claimed		

[Modify the Data Link Layer section to move Supports configuration through Network Port object to the Base Req.]

Data Link Layer - Secure Connect		
	R	Base Requirements
	C <sup>1</sup>	Is able to operate as a node without a local hub function
	C <sup>1</sup>	Is able to operate as a hub
	O	Supports direct connections
	O <sup>2</sup>	Is able to accept direct connections
	O <sup>2</sup>	Is able to initiate direct connections
	O	Supports configuration through Network Port object
	C <sup>3</sup>	<i>Supports Procedure to Replace BACnet/SC Certificates</i>
<sup>1</sup> At least one of these options must be supported. <sup>2</sup> At least one of these options must be supported if the device supports direct connections. <sup>3</sup> <i>required for devices claiming Protocol Revision 24 or higher</i>		

## Test Plan Changes

- [In BTL Test Plan, modify all references to 12.5.2.2.2 from 135.1-2023 to BTL]
- [In BTL Test Plan, modify all references to 12.5.3.2.2.1 from 135.1-2023 to BTL]
- [In BTL Test Plan, modify all references to 7.3.2.46.3.9 from 135.1-2023 to BTL]
- [In BTL Test Plan modify all references to 12.5.1.1.16 from 135.1-2023 to BTL]
- [In BTL Test Plan modify all references to 12.5.1.1.17 from 135.1-2023 to BTL]
- [In BTL Test Plan modify all references to 12.5.1.2.5 from 135.1-2023 to BTL]
- [In BTL Test Plan modify all references to 12.5.2.2.2 from 135.1-2023 to BTL]
- [In BTL Test Plan modify all references to 12.5.3.2.1 from 135.1-2023 to BTL]

[Insert the following section into 3.56 after 3.56.3]

### 3.56.4 Supports Non-hierarchical Network Port Objects

The IUT contains, or can be made to contain, non-hierarchical Network Port objects.

<b>Verify EPICS</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Verify IUT contains only Network Port objects with Protocol_Level equal to BACNET_APPLICATION or NON_BACNET_APPLICATION.
	<b>Testing Hints</b>	
<b>Verify EPICS</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Verify the Network Port object, Reference_Port and Additional_Reference_Ports properties are absent.
	<b>Testing Hints</b>	
<b>Verify EPICS</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT claims Protocol_Revision >= 24.
	<b>Test Directives</b>	Verify each Network Port object contains all required properties based on its Network_Type.
	<b>Testing Hints</b>	
<b>Verify EPICS</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT claims Protocol_Revision >= 24.
	<b>Test Directives</b>	Verify each Network Port object contains only valid optional properties based on its Network_Type.
	<b>Testing Hints</b>	

[Renumber all remaining entries in section 3.56]

[Modify section 3.56.4 (now renumbered to 3.56.5) as shown in yellow]

### 3.56.4.5 Supports Hierarchical Network Port Objects

The IUT contains, or can be made to contain, a set of Network Port objects which form a hierarchy of protocols.

<b>Verify EPICS</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT claims Protocol_Revision >= 24.
	<b>Test Directives</b>	Verify the IUT contains a Network Port object for each Protocol_Level based on the Network_Type.
	<b>Testing Hints</b>	
<b>Verify EPICS</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT claims Protocol_Revision >= 24.
	<b>Test Directives</b>	Verify that each Network Port object contains only required and optional properties based on its Network_Type and Protocol_Level.
	<b>Testing Hints</b>	
<b>BTL135.1-2023 - 7.3.2.46.4.1 - Valid Hierarchy Test</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Repeat for each supported Network_Type at the BACNET_APPLICATION level.
	<b>Testing Hints</b>	
<b>135.1-2023 - 7.3.2.46.4.2 - Properties in Referenced Network Port Reflected in Top Network Port Object</b>		
	<b>Test Conditionality</b>	Must be executed <i>if the IUT claims Protocol_Revision &lt; 24.</i>
	<b>Test Directives</b>	Repeat for each supported Network_Type at the BACNET_APPLICATION level.
	<b>Testing Hints</b>	The test is written such that it tests all configured BACNET_APPLICATION Network Port objects so configuring the IUT to contain an example of each will allow the test to be run fewer times.
<b>135.1-2023 - 7.3.2.46.4.3 - Changes Reflected in Top Network Port Object</b>		
	<b>Test Conditionality</b>	<i>Must be executed if the IUT claims Protocol_Revision &lt; 24 and supports writable</i> Test shall be skipped if the IUT does not support any writable properties in its Network Port hierarchies.
	<b>Test Directives</b>	Repeat for each supported Network_Type at the BACNET_APPLICATION level.
	<b>Testing Hints</b>	The test is written such that it tests all configured BACNET_APPLICATION Network Port objects so configuring the IUT to contain an example of each will allow the test to be run fewer times.
<b>135.1-2023 - 7.3.2.46.4.4 - Changes Reflected in Lower Network Port Objects</b>		
	<b>Test Conditionality</b>	<i>Must be executed if the IUT claims Protocol_Revision &lt; 24 and supports</i> Test shall be skipped if the IUT does not support any writable <i>writable</i> properties in its Network Port hierarchies.
	<b>Test Directives</b>	Repeat for each supported Network_Type at the BACNET_APPLICATION level.
	<b>Testing Hints</b>	The test is written such that it tests all configured BACNET_APPLICATION Network Port objects so configuring the IUT to contain an example of each will allow the test to be run fewer times.

[Insert two new section into 3.56 before the last section of Supports the Routing\_Table Property]

### 3.56.14 Supports the GENERATE\_CSR\_FILE Command

The IUT supports the GENERATE\_CSR\_FILE Command in Network Port objects.

BTL - 7.3.2.46.3.X.1 - GENERATE_CSR_FILE Command Test	
Test Conditionality	Must be executed if the IUT supports the GENERATE_CSR_FILE command.
Test Directives	
Testing Hints	
BTL - 7.3.2.46.3.X.2 - GENERATE_CSR_FILE Command Failure Test	
Test Conditionality	Must be executed if the IUT supports a Network Port object for which GENERATE_CSR_FILE is not supported.
Test Directives	
Testing Hints	

### 3.56.15 Supports the VALIDATE\_CHANGES Command

The IUT supports the VALIDATE\_CHANGES command in Network Port objects.

BTL - 7.3.2.46.3.X.3 - VALIDATE_CHANGES Command Test	
Test Conditionality	Must be executed if the IUT supports the VALIDATE_CHANGES command.
Test Directives	
Testing Hints	
BTL - 7.3.2.46.3.X.4 - VALIDATE_CHANGES Command Failure Test	
Test Conditionality	Must be executed if the IUT supports a Network Port object for which VALIDATE_CHANGES is not supported.
Test Directives	
Testing Hints	

[Renumber remaining section to 3.56.16]

[Add new section to 3.61 after section 3.61.2]

### 3.61.3 Supports BACnet/SC Certificate Exchange

The IUT supports BACnet/SC Certificate Exchange using AtomicReadFile and AtomicWriteFile requests.

Verify Checklist	
Test Conditionality	Must be executed.
Test Directives	Verify that the IUT claims support for section 9.9.8 Supports Procedure to Replace BACnet/SC Certificates.
Testing Hints	

[Renumber remaining sections in 3.61]

[Modify the title of the following three sections in 3.61]

### 3.61.3.3.61.4 Supports a Record-Based File Object for a Purpose Other Than Backup and Restore or BACnet/SC Certificate Exchange

For a device which contains a record-based File object for a purpose other than Backup and Restore or BACnet/SC Certificate Exchange, there are no testing requirements.

### 3.61.4.3.61.5 Supports a Stream-Based File Object for a Purpose Other Than Backup and Restore or BACnet/SC Certificate Exchange

The IUT supports a data stream-based File that is not accessed during Backup and Restore or BACnet/SC Certificate Exchange.

### 3.6.53.6.6 Contains a Writable Stream-Based File for a Purpose Other Than Backup and Restore *or BACnet/SC Certificate Exchange*

The IUT supports a data stream-based File that is not accessed during Backup and Restore *or BACnet/SC Certificate Exchange*.

<b>135.1-2023 - 9.13.1.2.1 - Writing an Entire Stream-Based File</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Apply to a file not used for Backup & Restore <i>or BACnet/SC Certificate Exchange</i> .
	<b>Testing Hints</b>	
<b>135.1-2023 - 9.13.1.2.3 - Appending Data to the End of a File</b>		
	<b>Test Conditionality</b>	If the file size cannot be changed or if the IUT does not support files that cannot be modified except by replacing the entire file, then this test shall be skipped.
	<b>Test Directives</b>	Apply to a file not used for Backup & Restore <i>or BACnet/SC Certificate Exchange</i> .
	<b>Testing Hints</b>	
<b>135.1-2023 - 9.13.1.2.4 - Truncating a File</b>		
	<b>Test Conditionality</b>	If the only value that the IUT accepts when writing File_Size is zero, then this test shall be skipped.
	<b>Test Directives</b>	Apply to a file not used for Backup & Restore <i>or BACnet/SC Certificate Exchange</i> .
	<b>Testing Hints</b>	
<b>135.1-2023 - 9.13.1.2.5 - Deleting a File</b>		
	<b>Test Conditionality</b>	If the file size cannot be changed, then this test shall be skipped.
	<b>Test Directives</b>	Apply to a file not used for Backup & Restore <i>or BACnet/SC Certificate Exchange</i> .
	<b>Testing Hints</b>	
<b>135.1-2023 - 9.13.2.2.1 - Writing to a Stream Access File using Record Access</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Apply to a file not used for Backup & Restore <i>or BACnet/SC Certificate Exchange</i> .
	<b>Testing Hints</b>	
<b>135.1-2023 - 9.13.2.2.2 - Writing to a File with an Invalid Starting Position</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Apply to a file not used for Backup & Restore <i>or BACnet/SC Certificate Exchange</i> .
	<b>Testing Hints</b>	
<b>135.1-2023 - 9.13.2.2.4 - Writing to a Nonexistent File</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

[In section 9.9.1 Data Link Layer - Secure Connect Base Requirements, modify the Test Conditionality for test 12.5.1.1.17]

<b>BTL 135.1-2023 - 12.5.1.1.17 - Configurable Reconnect Timeout Test</b>		
	<b>Test Conditionality</b>	<del>If the IUT has a fixed reconnect timeout, this test shall be skipped.</del> <i>If the IUT claims Protocol Revision 24 or greater, this test must be executed. If the IUT claims Protocol Revision 23 or lower and has a fixed reconnect timeout, this test shall be skipped</i>
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

...

<b>135.1-2023 - 12.5.1.2.6 - HTTP 1.1 Fallback Test</b>		
	<b>Test Conditionality</b>	This test shall be executed if the IUT supports BACnet/SC over <del>HTTP 2</del> <i>or a later version than HTTP 1.1</i>
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

...

<b>135.1-2023 - 7.3.2.46.1.2 - Verify Network Configuration Through Network Port Object Test</b>	
<b>Test Conditionality</b>	If the device claims Protocol_Revision 16 or lower, this test shall be skipped.
<b>Test Directives</b>	Perform at least once. Repeat each time the network is reconfigured for a test. Execute this test at least once on each Network Port object that has Network_Type = SECURE_CONNECT.
<b>Testing Hints</b>	

### 9.9.7 Supports Configuration Through Network Port Object

The IUT supports full, or partial, configuration of the data link through the Network Port object. Specifically, at least 1 property in the Network Port object which changes the behavior of the data link is writable.

<b>135.1-2023 - 7.3.2.46.1.1 - Configure Network Through Network Port Object Test</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Perform at least once. Repeat each time the network is reconfigured for a test. Execute this test at least once on each Network Port object that has Network_Type = SECURE_CONNECT.
<b>Testing Hints</b>	

### 9.9.8 Supports Procedure to Replace BACnet/SC Certificates

The IUT supports full Procedure to Replace BACnet/SC Certificates. BACnet/SC certificates will usually have a finite lifetime. A procedure to replace expiring certificates is required to ensure the network is minimally interrupted during this transition.

<b>Verify Checklist</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Verify that the IUT claims support for section 3.61.3 Supports BACnet/SC Certificate Exchange
<b>Testing Hints</b>	
<b>BTL - 19.Y.3.1 Adding a New Issuer Certificate to the Device</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	
<b>BTL - 19.Y.3.2 Replace the Operational Certificate</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	
<b>BTL - 19.Y.3.3 Removing an Outdated Issuer Certificate from the Device</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	

---

## Specified Test Changes

---

[Move test from 135.1 into BTL with changes as marked.]

### 7.3.2.46.4.1 Valid Hierarchy Test

Reason for Change: Modified per Addendum 135-2020cc-1.

Purpose: To verify that the set of network port objects in the IUT are organized in a valid hierarchy.

Test Concept: Visit each Network Port object which represents a configured application layer port. Ensure that the top Network Port object has a Protocol\_Level of BACNET\_APPLICATION or NON\_BACNET\_APPLICATION. Visit each Network Port object in the hierarchy ensuring that the Protocol\_Level properties are valid.

```

1. REPEAT NP = (object id of each hierarchical Network Port object which has a Protocol_Level of
    BACNET_APPLICATION or NON_BACNET_APPLICATION) {
    REPEAT NPx = (object id of each Network Port object , Reference_Port in NP's hierarchy) {
        PL = READ (Network Port, NPx), Protocol_Level
        IF PL is BACNET_APPLICATION or NON_BACNET_APPLICATION THEN
            ERROR Invalid Protocol_Level in child Network Port object
        IF PL is PHYSICAL THEN
            VERIFY (Network Port, NPx), Reference_Port = 4194303
    }
    IF (Protocol_Revision >= 24 and Additional_Reference_Ports is present) THEN
        IF (NP, Reference_Port property is not present) THEN
            ERROR missing Reference_Port property
        REPEAT (for each entry Network Port object, Additional_Reference_Ports) DO {
            REPEAT NPx = (object id of each Network Port object, Additional_Reference_Ports in NP's hierarchy)
DO {
                PL = READ (Network Port, NPx), Protocol_Level
                IF PL is BACNET_APPLICATION or NON_BACNET_APPLICATION THEN
                    ERROR Invalid Protocol_Level in child Network Port object
                IF PL is PHYSICAL THEN
                    VERIFY (Network Port, NPx), Additional_Reference_Ports = (empty list)
            }
        }
    }
}

```

[Move test from 135.1 into BTL Specified Tests with changes marked.]

### 7.3.2.46.3.9 No Commands if Changes\_Pending Test

Reason for Change: Modified per Addendum 135-2020cc-1.

Purpose: To verify that the Network Port disallows commands, except DISCARD\_CHANGES and VALIDATE\_CHANGES, when Changes\_Pending.

Test Concept: using Network Port object NP, write values to one or more properties, P1 .. Px, which utilize the pending changes functionality. Write each of the other commands and verify they are rejected.

Configuration Requirements: Execute the test on a Network Port object which supports the Command property.

```

-- write some properties
1. REPEAT P = (P1 .. Px) {
    WRITE NP, P = (any valid value)
}

-- verify that changes are pending
2. VERIFY Changes_Pending = TRUE

-- write each supported Command value, except DISCARD_CHANGES and VALIDATE_CHANGES
3. REPEAT CMD = (all non-IDLE valid values that NP supports except DISCARD_CHANGES and
VALIDATE_CHANGES) {
    TRANSMIT WriteProperty-Request
        'Object Identifier' = NP
        'Property' = Command,

```

```

    'Property Value' = CMD
    RECEIVE BACnet-Error-PDU
    'Error Class' =          PROPERTY,
    'Error Code' =          INVALID_VALUE_IN_THIS_STATE
}

```

-- revert the Network Port object

```

4. IF the IUT supports DISCARD_CHANGES THEN {
    WRITE Command = DISCARD_CHANGES
} ELSE {
    MAKE (the IUT discard its changes)
}

```

[Add new test to BTL Specified Tests]

### ~~7.3.2.43.3.X.1 GENERATE\_CSR\_FILE Command Test~~

~~Reason for Change: New test per Addendum 135-2020cc-1.~~

~~Purpose: To verify that the Network Port attempts to generate a new csr\_file when commanded to.~~

~~Test Concept: Starting with a Network Port object which supports the GENERATE\_CSR\_FILE command. The port is commanded to GENERATE\_CSR\_FILE. The new certificate signing request file shall be referenced from the Certificate\_Signing\_Request\_File property. The certificate signing request file shall be updated and a new private/public key pair shall be generated when the Command property is written to GENERATE\_CSR\_FILE. The certificate signing request file shall contain the X.509 'subject' distinguished name specified in the currently active operational certificate, and the public key from the new private/public key pair. If no operational certificate is currently active, then the X.509 'subject' distinguished name of the certificate signing request is a local matter but shall be globally unique.~~

~~Configuration Requirements: Execute the test on a Network Port object which supports the Command property and property Changes Pending = FALSE~~

~~–CSR = This read only property, of type BACnetObjectIdentifier, specifies the File object that contains the PKCS #10 certificate signing request as defined by RFC 5967 in PEM format.~~

~~Test Steps:~~

```

1. READ CSR = Certificate_Signing_Request_File
2. CHECK (CSR is non empty BACnetObjectIdentifier referring to File object)
3. WHILE (the last read did not indicate 'End Of File') DO {
    _____ TRANSMIT AtomicReadFile Request,
    _____ 'Object Identifier' = CSR,
    _____ 'File Start Position' = (the next unread octet),
    _____ 'Requested Octet Count' = MROC
    _____ RECEIVE AtomicReadFile ACK,
    _____ 'End Of File' = TRUE | FALSE,
    _____ 'File Start Position' = (the next unread octet)
    _____ 'File Data' = (CSR File contents of length MROC if 'End Of File' is FALSE
    _____ or of length MROC or less if 'End Of File' is TRUE)
    _____ }
4. TRANSMIT WriteProperty Request,
   'Object Identifier' = _____ (the Network Port object),
   'Property Identifier' = _____ Command,
   'Property Value' = _____ GENERATE_CSR_FILE
5. RECEIVE BACnet SimpleACK
6. VERIFY Changes_Pending = FALSE
7. WHILE (Command <=> IDLE) DO {}
8. READ CSR = Certificate_Signing_Request_File
9. CHECK (CSR is non empty BACnetObjectIdentifier referring to File object)
10. WHILE (the last read did not indicate 'End Of File') DO {
    _____ TRANSMIT AtomicReadFile Request,
    _____ 'Object Identifier' = CSR,
    _____ 'File Start Position' = (the next unread octet),

```

```


'Requested Octet Count' = MROC
RECEIVE AtomicReadFile_ACK,
'End Of File' = TRUE | FALSE,
'File Start Position' = (the next unread octet)
'File Data' = (CSR File contents of length MROC if 'End Of File' is FALSE
or of length MROC or less if 'End Of File' is TRUE)
}


```

11. ~~VERIFY (that a file content from Step 3 <> file content from Step 10)~~

### 7.3.2.46.3.X.1 GENERATE\_CSR\_FILE Command Test

Reason for Change: New test per Addendum 135-2020cc-1. Added File object File\_Size and Modification\_Date checks.

Purpose: To verify that the Network Port object generates a new CSR file when commanded to.

Test Concept: Using a Network Port object, NP1, which supports the GENERATE\_CSR\_FILE command, the port is commanded to GENERATE\_CSR\_FILE. Test the referenced CSR file has been updated using the Modification\_Date property.

Configuration Requirements: Execute the test on a Network Port object which supports the Command property and property Changes\_Pending = FALSE.

Test Steps:

1. WRITE NP1, Command = GENERATE\_CSR\_FILE
2. WHILE (NP1, Command <> IDLE) DO {}
3. VERIFY Changes\_Pending = FALSE
4. READ CSR = NP1, Certificate\_Signing\_Request\_File
5. VERIFY CSR, Modification\_Date = (the current local date and time)
6. VERIFY CSR, File\_Size > 0

[Add new test to BTL Specified Tests]

### 7.3.2.46.3.X.2 GENERATE\_CSR\_FILE Command Failure Test

Reason for Change: New test per Addendum 135-2020cc-1.

Purpose: To verify that Network Port objects respond to the GENERATE\_CSR\_FILE command with the correct error codes when the command is not supported / enabled.

Test Concept: With a Network Port object for a network which does not support GENERATE\_CSR\_FILE. Verify that the correct error code is returned.

Configuration Requirements: property Changes Pending = FALSE.

1. TRANSMIT WriteProperty-Request,
  - 'Object Identifier' = (the Network Port object),
  - 'Property Identifier' = Command,
  - 'Property Value' = GENERATE\_CSR\_FILE
2. RECEIVE BACnet-Error-PDU
  - 'Error Class' = PROPERTY,
  - 'Error Code' = OPTIONAL\_FUNCTIONALITY\_NOT\_SUPPORTED

[Add new test to BTL Specified Tests]

### 7.3.2.46.3.X.3 VALIDATE\_CHANGES Command Test

Reason for Change: New test per Addendum 135-2020cc-1.

Purpose: To verify that the Network Port attempts to perform the required validations on this property when commanded to.

Test Concept: Starting with a Network Port object which supports the VALIDATE\_CHANGES command. The port is commanded to VALIDATE\_CHANGES. This command shall initiate a validation of the values of the properties of this port as specified in each property. If a property is present but not used, based on the Network\_Type, it shall not be validated. The value of the Command\_Validation\_Result property shall be updated to indicate the validation result.

1. READ V1 = Command\_Validation\_Result
2. READ CP = Changes\_Pending
- request a VALIDATE\_CHANGES command, and wait for it to timeout
3. WRITE Command = VALIDATE\_CHANGES
4. VERIFY Changes\_Pending = CP
5. WHILE (Command <> IDLE) DO {}
6. VERIFY Command\_Validation\_Result = Any value different from V1

[Add new test to BTL Specified Tests]

### 7.3.2.46.3.X.4 VALIDATE\_CHANGES Command Failure Test

Reason for Change: New test per Addendum 135-2020cc-1.

Purpose: To verify that Network Port objects respond to the VALIDATE\_CHANGES command with the correct error codes when the command is not supported / enabled.

Test Concept: With a Network Port object for a network which does not support VALIDATE\_CHANGES. Verify that the correct error code is returned.

1. TRANSMIT WriteProperty-Request,
  - 'Object Identifier' = (the Network Port object),
  - 'Property Identifier' = Command,
  - 'Property Value' = VALIDATE\_CHANGES
2. RECEIVE BACnet-Error-PDU
  - 'Error Class' = PROPERTY,
  - 'Error Code' = OPTIONAL\_FUNCTIONALITY\_NOT\_SUPPORTED

[Copy this test from 135.1-2023 and put into BTL Specified tests with the marked changes]

### 12.5.1.1.16 Heartbeat-Request Initiation Test

Reason for Change: modified per Addendum 135-2020cc-1.

Purpose: To verify that the device initiates heartbeats as per its config.

Test Concept: With the IUT connected to the BACnet/SC network, *send a ReadProperty request to the IUT every heartbeat interval / 2 seconds. Verify that the IUT does not initiate a Heartbeat Request. Stop sending messages to the IUT. Wait the IUT's configured heart-beat interval plus 10 seconds and verify that the IUT sent a Heartbeat-Request, ensuring that no BVLCs are sent to the IUT during that period. If the IUT claims Protocol\_Revision 24 or greater heartbeat interval is the Network Port object, SC\_Heartbeat\_Timeout property.*

Configuration Requirements: Place the IUT in a mode where it will not initiate requests for a period longer than the heartbeat interval (except for the heartbeat request). If the IUT does not support DM-DCC-B and cannot be otherwise configured to behave in this manner, this test shall be skipped.

Test Steps:

1. REPEAT N = (1..Z) {
  - TRANSMIT Encapsulated-NPDU,
  - 'Message ID' = (M: any valid value),

```

'Originating Virtual Address' =      (OVA: any valid value, including absent),
-- 'Destination Virtual Address' absent
'Destination Options' =              (absent or any valid value),
'Data Options' =                    ({ X'41' }), -- Secure Path
'BACnet NPDU' =
    ReadProperty-Request,
    'Object Identifier' =            (the IUT's Device object),
    'Property Identifier' =          Object_Name
RECEIVE Encapsulated-NPDU,
'Message ID' =                      M,
-- 'Originating Virtual Address' absent
'Destination Virtual Address' =      OVA,
'Destination Options' =              (absent or any valid value),
'Data Options' =                    ({ X'41' or a list of valid header options including Secure
Path}),
'BACnet NPDU' =
    ReadProperty-ACK,
    'Object Identifier' =            (the IUT's Device object),
    'Property Identifier' =          Object_Name,
    'Property Value' =              (the IUT's device object name)
}
2. WAIT 1/2 of IUT's heartbeat interval
}
2. CHECK(that the IUT did not send a HeartBeat during step 1)
}
Since we already waited 1/2 of an heartbeat interval, only 1/2 of that interval is now given for the IUT to generate a Heartbeat Request
23. BEFORE 1/2 of IUT's heartbeat interval + 10s
    RECEIVE Heartbeat-Request,
        'Message ID' =                M2,
        -- 'Originating Virtual Address' absent
        -- 'Destination Virtual Address' absent
        'Destination Options' =        (absent or any valid value),
        -- 'Data Options' absent
34. TRANSMIT Heartbeat-ACK,
        'Message ID' =                M2,
        -- 'Originating Virtual Address' absent
        -- 'Destination Virtual Address' absent
        'Destination Options' =        (absent or any valid value),
        -- 'Data Options' absent

```

[Copy this test from 135.1-2023 and put into BTL Specified tests with the marked changes]

### 12.5.1.1.17 Configurable Reconnect Timeout Test

Reason for Change: modified per Addendum 135-2020cc-1.

Purpose: To verify that a device adheres to its configurable reconnect timeout.

Test Concept: Turn on the IUT. When the IUT attempts to connect to the primary hub, the primary hub does not respond. Verify that the IUT waits at least the configured reconnect timeout, *minRT*, and no longer than *maxRT* seconds before attempting to reconnect. *If the IUT claims Protocol Revision 23 or lower, minRT is a configurable parameter within the IUT and maxRT is fixed at 600 seconds. If the IUT claims Protocol Revision 24 or greater, minRT is the Network Port object, SC Minimum Reconnect Time property and maxRT is the SC Maximum Reconnect Time property.*

Configuration Requirements: The IUT is configured with the TD as the primary hub with no failover hub or as direct connection initiation peer of the TD. The IUT is configured with a tester selected reconnect timeout, *minRT*, within the range supported by the IUT and within 2 .. 300 seconds and *maxRT*, within the range supported by the IUT and within 2 .. 600 seconds. The IUT starts the test *disconnected from the TD* powered off. *If the IUT has a fixed reconnect timeout, this test shall be skipped.*

Test Steps:

1. MAKE(place the TD in a mode where it will not accept a websocket connectionincoming connections)
24. MAKE(the IUT attempt to connect to the TD)
3. T1 = Local Time
2. CHECK(that the IUT attempts to open a new WebSocket with the TD)
43. MAKE(place the TD in a mode where it will accept incoming connections)
4. WAIT minRT seconds
5. BEFORE 600 RT seconds
  - RECEIVE PORT (IUT-TD primary hub WebSocket)
    - Connect-Request,
    - 'Message ID' = (M1: any valid value),
    - 'Originating Virtual Address' absent
    - 'Destination Virtual Address' absent
    - 'Destination Options' (absent or any valid value),
    - 'Data Options' absent
    - 'VMAC Address' = (IUT's VMAC),
    - 'Device UUID' = (IUT's UUID),
    - 'Maximum BVLC Length' = (the IUT's maximum BVLC accepted length),
    - 'Maximum NPDU Length' = (the IUT's maximum NPDU accepted length)
6. TRANSMIT PORT (IUT-TD primary hub WebSocket)
  - Connect-Accept,
  - 'Message ID' = M1,
  - 'Originating Virtual Address' absent
  - 'Destination Virtual Address' absent
  - 'Destination Options' (absent or any valid value),
  - 'Data Options' absent
  - 'VMAC Address' = (TD's VMAC),
  - 'Device UUID' = (TD's UUID),
  - 'Maximum BVLC Length' = (the TD's maximum BVLC accepted length),
  - 'Maximum NPDU Length' = (the TD's maximum NPDU accepted length)
7. T2 = Local Time
8. CHECK (T2 - T1 >= minRT)
9. CHECK (T2 - T1 <= maxRT)

[Copy this test from 135.1-2023 and put into BTL Specified tests with the marked changes]

### 12.5.1.2.5 Connect-Request Response Wait Time Test

Reason for Change: modified per Addendum 135-2020cc-1.

Purpose: To verify that the device will close the WebSocket if a response to a Connect-Request is not received before the connection wait timer expires. *If the IUT claims Protocol Revision 24 or greater connect wait timeout is the Network Port object, SC Connect Wait Timeout property.*

Test Concept: Turn on the IUT. When the IUT attempts to connect to the TD as the primary hub or as a direct connection peer, the TD will accept the WebSocket connection but will not send a response to the connect request. It is verified that the IUT closes the WebSocket when the connection wait timer expires.

Configuration Requirements: The IUT is configured with the TD as the primary hub, or as a direct connect peer. The TD is configured to accept WebSocket connections but to not respond to Connect-Requests.

Test Steps:

1. MAKE(the IUT connect to the TD)
2. CHECK(that the IUT attempts to open a new WebSocket with the TD)
3. RECEIVE Connect-Request,
  - 'Message ID' = (M1: any valid value),
  - 'Originating Virtual Address' absent
  - 'Destination Virtual Address' absent
  - 'Destination Options' (absent or any valid value),
  - 'Data Options' absent
  - 'VMAC Address' = (IUT's VMAC),

- 'Device UUID' = (IUT's UUID),
  - 'Maximum BVLC Length' = (the IUT's maximum BVLC accepted length),
  - 'Maximum NPDU Length' = (the IUT's maximum NPDU accepted length)
4. WAIT connect wait timeout
  5. CHECK(that the IUT closed the WebSocket)

[Copy this test from 135.1-2023 and put into BTL Specified tests with the marked changes]

#### 12.5.2.2.2 Connect-Request Wait Time Test

Reason for Change: modified per Addendum 135-2020cc-1.

Purpose: To verify that the hub will close the WebSocket if the Connect-Request is not received before the connection wait timer expires.

Test Concept: With the IUT connected to the BACnet/SC network. Open a WebSocket connection with the IUT's hub port, but do not send a connect-request. Verify that the IUT closes the WebSocket after the connection wait timer expires. *If the IUT claims Protocol\_Revision 24 or greater connect wait timeout is the Network Port object, SC\_Connect\_Wait\_Timeout property.*

Configuration Requirements: The IUT is configured to be a BACnet/SC hub.

Test Steps:

1. MAKE(a WebSocket connection to the IUT's hub function)
2. WAIT the connection wait timer expiration time
3. CHECK(that the IUT closed the WebSocket and did not send any messages on the WebSocket)

[Copy this test from 135.1-2023 and put into BTL Specified tests with the marked changes]

#### 12.5.3.2.2.1 Connect-Request Wait Time Test

Reason for Change: Updated to include SC\_Connect\_Wait\_Timeout property if available.

Purpose: To verify that the IUT will close the WebSocket if the Connect-Request is not received before the connection wait timer expires.

Test Concept: With the IUT connected to the BACnet/SC network, open a WebSocket connection to the IUT's direct connect URI, but do not send a connect-request. Verify that the IUT closes the WebSocket after the connection wait timer expires. *If the IUT claims Protocol\_Revision 24 or greater connection wait timer is the Network Port object, SC\_Connect\_Wait\_Timeout property.*

Configuration Requirements: The IUT is configured to accept direct connections.

Test Steps:

1. MAKE(a WebSocket connection to the IUT's direct connect WebSocket-URI)
2. WAIT the connection wait timer expiration time
3. CHECK(that the IUT closed the WebSocket and did not send any messages on the WebSocket)



```


TRANSMIT AtomicWriteFile-Request
  'File Identifier' = IC[2]
  'File Start Position' = Z
  'Record Data' = (file contents of the new issuer certificate, the number of octets
being the lesser of (file size - Z) and MWDL)
RECEIVE AtomicWriteFile-ACK
  'File Start Position' = Z
}
ELSE

READ FS = IC[1], File_Size
WRITE IC[1], File_Size = 0
REPEAT Z = (0 through the file size, in increments of MWDL) DO {
  TRANSMIT AtomicWriteFile-Request
  'File Identifier' = IC[1]
  'File Start Position' = Z
  'Record Data' = (file contents of the new issuer certificate, the number of octets
being the lesser of (file size - Z) and MWDL)
  RECEIVE AtomicWriteFile-ACK
  'File Start Position' = Z
}
}
}
}

6. TRANSMIT ReinitializeDevice-Request
  'Reinitialized State of Device' = WARMSTART | ACTIVATE_CHANGES
  'Password' = (any valid password)
7. RECEIVE BACnet-SimpleACK-PDU
8. WAIT Activate Changes Fail Time
9. VERIFY Changes_Pending = FALSE
10. VERIFY the connection shall be re-established.


```

### 19.Y.3.1 Adding a New Issuer Certificate to the Device

New test per Addendum 135-2020cc-3. Added File object File\_Size and Modification\_Date checks.

Purpose: This test verifies the IUT can execute the procedure to add a new Issuer Certificate.

Test Concept: With the IUT connected to a BACnet/SC network, this test takes the IUT through the procedure to add a new Issuer Certificate to a file object referenced in the Network Port object, NP1, of the IUT of the active BACnet/SC network. The procedure is specified in 135-2020 Addendum cc, Clause 19.Y.3.1.

Configuration Requirements: The IUT is actively connected to a BACnet/SC network.

Notes To Tester: When performing the AtomicReadFile and AtomicWriteFile services, a Maximum Requested Octet Count (MROC) and a Maximum Write Data Length (MWDL) shall be calculated before starting the test. These values shall be used during the test. MROC shall be 16 octets less than the minimum of the TD's Max\_APDU\_Length\_Accepted and the IUT's maximum transmittable APDU length. MWDL shall be 21 octets less than the minimum of the TD's maximum transmittable APDU length and the IUT's Max\_APDU\_Length\_Accepted. (See 135-2020 Addendum cc, Clause 19.Y.3.1)

Test Steps:

1. READ NP1, IC = (Issuer\_Certificate\_Files 1, Issuer\_Certificate\_Files 2)
2. FS = READ (IC[1], File\_Size)
3. IF (FS > 0) THEN {
  - WHILE (the last read did not indicate 'End Of File') DO {
    - TRANSMIT AtomicReadFile-Request,
    - 'Object Identifier' = IC[1],
    - 'File Start Position' = (the next unread octet),
    - 'Requested Octet Count' = MROC
  - RECEIVE AtomicReadFile-ACK,
  - 'End Of File' = TRUE | FALSE,

```

        'File Start Position' = (the next unread octet)
        'File Data' = (IC[1] File contents of length MROC if 'End Of File' is FALSE
        or of length MROC or less if 'End Of File' is TRUE)
    }
}
4 FS = READ (IC[2], File_Size)
5. IF (FS > 0) THEN {
    WHILE (the last read did not indicate 'End Of File') DO {
        TRANSMIT AtomicReadFile-Request,
        'Object Identifier' = IC[2],
        'File Start Position' = (the next unread octet),
        'Requested Octet Count' = MROC
        RECEIVE AtomicReadFile-ACK,
        'End Of File' = TRUE | FALSE,
        'File Start Position' = (the next unread octet)
        'File Data' = (IC[2] File contents of length MROC if 'End Of File' is FALSE
        or of length MROC or less if 'End Of File' is TRUE)
    }
}
6. IF (IC[1] File contents is = the IUT's known current issuer certificate) THEN {
    WRITE IC[2], File_Size = 0
    VERIFY IC[2], Modification_Date = (the current local data and time)
    VERIFY NP1, Changes_Pending = TRUE
    REPEAT Z = (0 through the file size, in increments of MWDL) DO {
        TRANSMIT AtomicWriteFile-Request
        'File Identifier' = IC[2]
        'File Start Position' = Z
        'Record Data' = (file contents of the new issuer certificate, the number of octets
        being the lesser of (file size - Z) and MWDL)
        RECEIVE AtomicWriteFile-ACK
        'File Start Position' = Z
    }
    WAIT Internal Processing Fail Time
    VERIFY IC[2], Modification_Date = (the current local data and time)
    VERIFY NP1, Changes_Pending = TRUE
}
ELSE
{
    WRITE IC[1], File_Size = 0
    VERIFY IC[1], Modification_Date = (the current local data and time)
    VERIFY NP1, Changes_Pending = TRUE
    REPEAT Z = (0 through the file size, in increments of MWDL) DO {
        TRANSMIT AtomicWriteFile-Request
        'File Identifier' = IC[1]
        'File Start Position' = Z
        'Record Data' = (file contents of the new issuer certificate, the number of octets
        being the lesser of (file size - Z) and MWDL)
        RECEIVE AtomicWriteFile-ACK
        'File Start Position' = Z
    }
    WAIT Internal Processing Fail Time
    VERIFY IC[1], Modification_Date = (the current local data and time)
    VERIFY NP1, Changes_Pending = TRUE
}
}
7. TRANSMIT ReinitializeDevice-Request
    'Reinitialized State of Device' = WARMSTART | ACTIVATE_CHANGES
    'Password' = (any valid password)
8. RECEIVE BACnet-SimpleACK-PDU
9. WAIT Activate Changes Fail Time
10. CHECK (the connection is re-established)

```

11. VERIFY NP1, Changes\_Pending = FALSE

**19.Y.3.2 Replace the Operational Certificate**

Reason for Change: New test per Addendum 135-2020cc 3.

Purpose: This test case verifies that the IUT can execute a replacement of operational certificate

Test Concept:

1. If a new private/public key pair is required and supported, device A writes the GENERATE\_CSR\_FILE command to the Command property.
2. Device A uploads the File object that contains the CSR from the node.
3. Device A requests the signing CA generate a new operational certificate based on the CSR data from step 2.
4. Device A writes the File object that represents the operational certificate for this node.
5. Device A sends a ReinitializeDevice(ACTIVATE\_CHANGES or WARMSTART, <password>) message to activate the changes to the Network Port object.

Configuration Requirements: These procedures assume that the BACnet/SC device to be configured contain a Network Port object with Network\_Type of SECURE\_CONNECT, the BACnet/SC network is active, this node have a valid operational certificate. When performing the AtomicReadFile and AtomicWriteFile services, a Maximum Requested Octet Count (MROC) and a Maximum Write Data Length (MWDL) shall be calculated before starting the test. These values shall be used during the test. MROC shall be 16 octets less than the minimum of the TD's Max\_APDU\_Length\_Accepted and the IUT's maximum transmittable APDU length. MWDL shall be 21 octets less than the minimum of the TD's maximum transmittable APDU length and the IUT's Max\_APDU\_Length\_Accepted. (See 135-2020 Addendum cc, Clause 19.Y.3.2)

Define:

CSR = Network Port object, Certificate\_Signing\_Request\_File property. See 135-2020 Addendum cc, Clause 12.56.Y26

OC = Network Port object, Operational\_Certificate\_File property. See 135-2020 Addendum cc, Clause 12.56.Y24

Test Steps:

1. IF (the IUT is capable of generating a certificate signing request file) THEN
  - \_\_\_\_\_ request the IUT generate a CSR, and wait for it to timeout
  - \_\_\_\_\_ WRITE Command = GENERATE\_CSR\_FILE
  - \_\_\_\_\_ WHILE (Command <> IDLE) DO {}
  - \_\_\_\_\_ ELSE
  - \_\_\_\_\_ MAKE (CSR File contains the PKCS #10 certificate signing request in PEM format)
2. READ CSR = Certificate\_Signing\_Request\_File
3. READ OC = Operational\_Certificate\_File
4. CHECK (CSR is non empty BACnetObjectIdentifier referring to File object)
5. WHILE (the last read did not indicate 'End Of File') DO {
  - \_\_\_\_\_ TRANSMIT AtomicReadFile Request,
  - \_\_\_\_\_ 'Object Identifier' = CSR,
  - \_\_\_\_\_ 'File Start Position' = (the next unread octet),
  - \_\_\_\_\_ 'Requested Octet Count' = MROC
  - \_\_\_\_\_ RECEIVE AtomicReadFile ACK,
  - \_\_\_\_\_ 'End Of File' = TRUE | FALSE,
  - \_\_\_\_\_ 'File Start Position' = (the next unread octet)
  - \_\_\_\_\_ 'File Data' = (CSR File contents of length MROC if 'End Of File' is FALSE
  - \_\_\_\_\_ or of length MROC or less if 'End Of File' is TRUE)
  - \_\_\_\_\_ }
  - \_\_\_\_\_ Requests the signing CA generate a new operational certificate OC Data based on the CSR CSR File contents data from step 5
6. READ FS = OC, File\_Size \_\_\_\_\_
  - \_\_\_\_\_ WRITE OC, File\_Size = 0
  - \_\_\_\_\_ REPEAT Z = (0 through the file size, in increments of MWDL) DO {
  - \_\_\_\_\_ TRANSMIT AtomicWriteFile Request

```

_____ 'File Identifier' = OC
_____ 'File Start Position' = Z
_____ 'Record Data' = (file contents of the new operational certificate OC
_____ Data, the number of octets
_____ being the lesser of (file size - Z) and MWDL)
_____ RECEIVE AtomicWriteFile ACK
_____ 'File Start Position' = Z
_____ }
7. TRANSMIT ReinitializeDevice Request
_____ 'Reinitialized State of Device' = WARMSTART | ACTIVATE_CHANGES
_____ 'Password' = (any valid password)
8. RECEIVE BACnet SimpleACK PDU
9. WAIT Activate Changes Fail Time
10. VERIFY Changes_Pending = FALSE
11. VERIFY the connection shall be re-established.

```

### 19.Y.3.2 Replace the Operational Certificate

New test per *Addendum 135-2020cc-3*. Added File object File\_Size and Modification\_Date checks.

Purpose: This test verifies the IUT can execute the procedure to replace the operational certificate.

Test Concept: With the IUT connected to a BACnet/SC network, this test takes the IUT through the procedure to replace the Operational Certificate to a file object referenced in the Network Port object, NP1, of the active BACnet/SC network. The procedure is specified in 135-2020 Addendum cc, Clause 19.Y.3.2.

Configuration Requirements: The IUT is actively connected to the BACnet/SC network.

Notes To Tester: When performing the AtomicReadFile and AtomicWriteFile services, a Maximum Requested Octet Count (MROC) and a Maximum Write Data Length (MWDL) shall be calculated before starting the test. These values shall be used during the test. MROC shall be 16 octets less than the minimum of the TD's Max\_APDU\_Length\_Accepted and the IUT's maximum transmittable APDU length. MWDL shall be 21 octets less than the minimum of the TD's maximum transmittable APDU length and the IUT's Max\_APDU\_Length\_Accepted.

Test Steps:

1. IF (the IUT is capable of generating a certificate signing request file) THEN
  - request the IUT generate a CSR, and wait for it to timeout
  - WRITE NP1, Command = GENERATE\_CSR\_FILE
  - WHILE (NP1, Command <> IDLE) DO {}
  - ELSE
    - MAKE (The Certificate\_Signing\_Request\_File of NP1 contain a new CSR File)
2. READ CSR = NP1, Certificate\_Signing\_Request\_File
3. VERIFY CSR, File\_Size > 0
4. READ OC = NP1, Operational\_Certificate\_File
5. WHILE (the last read did not indicate 'End Of File') DO {
  - TRANSMIT AtomicReadFile-Request,
    - 'Object Identifier' = CSR,
    - 'File Start Position' = (the next unread octet),
    - 'Requested Octet Count' = MROC
  - RECEIVE AtomicReadFile-ACK,
    - 'End Of File' = TRUE | FALSE,
    - 'File Start Position' = (the next unread octet)
    - 'File Data' = (CSR File contents of length MROC if 'End Of File' is FALSE or of length MROC or less if 'End Of File' is TRUE)
6. MAKE (The signing CA generate a new operational certificate based on the CSR File)
7. WRITE OC, File\_Size = 0
8. VERIFY OC, Modification\_Date = (the current local data and time)
9. VERIFY NP1, Changes\_Pending = TRUE
10. REPEAT Z = (0 through the file size, in increments of MWDL) DO {

```

TRANSMIT AtomicWriteFile-Request
  'File Identifier' = OC
  'File Start Position' = Z
  'Record Data' = (file contents of the new operational certificate,
    the number of octets being the lesser of (file size - Z) and MWDL)
RECEIVE AtomicWriteFile-ACK
  'File Start Position' = Z
}

```

11. WAIT Internal Processing Fail Time
11. VERIFY OC, File\_Size > 0
12. VERIFY OC, Modification\_Date = (the current local data and time)
13. VERIFY NP1, Changes\_Pending = TRUE
14. TRANSMIT ReinitializeDevice-Request
  - 'Reinitialized State of Device' = WARMSTART | ACTIVATE\_CHANGES
  - 'Password' = (any valid password)
15. RECEIVE BACnet-SimpleACK-PDU
16. WAIT Activate Changes Fail Time
17. CHECK (the connection is re-established)
18. VERIFY NP1, Changes\_Pending = FALSE

### 19.Y.3.3 Removing an Outdated Issuer Certificate from the Device

~~Reason for Change: New test per Addendum 135-2020cc-3.~~

~~Purpose: This test case verifies that the IUT can execute a Removing an Issuer Certificate.~~

~~Test Concept:~~

- ~~1. Device A determines the File object that represents the currently used issuer certificate for this network by reading the file content of one of the issuer certificate files and comparing it to the known current issuer certificate.~~
- ~~2. Device A writes zero to the File\_Size property of the issuer certificate File object that represents the other issuer certificate.~~
- ~~3. Device A sends a ReinitializeDevice(ACTIVATE\_CHANGES or WARMSTART, <password>) message to activate the changes to the Network Port object.~~

~~Configuration Requirements: These procedures assume that the BACnet/SC device is configured to contain a Network Port object with Network\_Type of SECURE\_CONNECT, the BACnet/SC network is active, this node have a valid operational certificate. When performing the AtomicReadFile and AtomicWriteFile services, a Maximum Requested Octet Count (MROC) and a Maximum Write Data Length (MWDL) shall be calculated before starting the test. These values shall be used during the test. MROC shall be 16 octets less than the minimum of the TD's Max\_APDU\_Length\_Accepted and the IUT's maximum transmittable APDU length. MWDL shall be 21 octets less than the minimum of the TD's maximum transmittable APDU length and the IUT's Max\_APDU\_Length\_Accepted. (See 135-2020 Addendum cc, Clause 19.Y.3.3)~~

~~Define:~~

~~IC = Network Port object, Issuer\_Certificate\_File property. See 135-2020 Addendum cc, Clause 12.56.Y25.~~

~~Test Steps:~~

- ~~1. READ IC = (Issuer\_Certificate\_Files 1, Issuer\_Certificate\_Files 2)~~
- ~~2. CHECK (CSR is non empty of type BACnetARRAY[2] of BACnetObjectIdentifier referring to 2 File objects)~~
- ~~3. WHILE (the last read did not indicate 'End Of File') DO {~~
  - ~~TRANSMIT AtomicReadFile Request,~~
  - ~~'Object Identifier' = IC[1],~~
  - ~~'File Start Position' = (the next unread octet),~~
  - ~~'Requested Octet Count' = MROC~~
  - ~~RECEIVE AtomicReadFile ACK,~~
  - ~~'End Of File' = TRUE | FALSE,~~
  - ~~'File Start Position' = (the next unread octet)~~
  - ~~'File Data' = (IC[1] File contents of length MROC if 'End Of File' is FALSE~~
    - ~~or of length MROC or less if 'End Of File' is TRUE)~~

```

4. WHILE (the last read did not indicate 'End Of File') DO {
   TRANSMIT AtomicReadFile-Request,
   'Object Identifier' = IC[2],
   'File Start Position' = (the next unread octet),
   'Requested Octet Count' = MROC
   RECEIVE AtomicReadFile-ACK,
   'End Of File' = TRUE | FALSE,
   'File Start Position' = (the next unread octet)
   'File Data' = (IC[2] File contents of length MROC if 'End Of File' is FALSE
   or of length MROC or less if 'End Of File' is TRUE)
   }
5. IF (IC[1] File contents is = the known current issuer certificate) THEN
   WRITE IC[2], File_Size = 0
   ELSE
   WRITE IC[1], File_Size = 0
6. TRANSMIT ReinitializeDevice-Request
   'Reinitialized State of Device' = WARMSTART | ACTIVATE_CHANGES
   'Password' = (any valid password)
7. RECEIVE BACnet-SimpleACK-PDU
8. MAKE (change the TD network setup and the network setup of all other devices on the network to match the target
   network setup)
9. WAIT Activate-Changes-Fail-Time
10. VERIFY Changes_Pending = FALSE


```

### 19.Y.3.3 Removing an Outdated Issuer Certificate from the Device

New test per Addendum 135-2020cc-3. Added File object File\_Size and Modification\_Date checks.

Purpose: This test verifies the IUT can execute the procedure to remove an Issuer Certificate.

Test Concept: With the IUT connected to a BACnet/SC network, this test takes the IUT through the procedure to remove an Issuer Certificate in the Network Port object, NP1, of the active BACnet/SC network. The procedure is specified in 135-2020 Addendum cc, Clause 19.Y.3.3.

Configuration Requirements: The IUT is actively connected to the BACnet/SC network and the NP1, Issuer\_Certificate\_Files property contains two Issuer Certificates.

Notes To Tester: When performing the AtomicReadFile and AtomicWriteFile services, a Maximum Requested Octet Count (MROC) and a Maximum Write Data Length (MWDL) shall be calculated before starting the test. These values shall be used during the test. MROC shall be 16 octets less than the minimum of the TD's Max\_APDU\_Length\_Accepted and the IUT's maximum transmittable APDU length. MWDL shall be 21 octets less than the minimum of the TD's maximum transmittable APDU length and the IUT's Max\_APDU\_Length\_Accepted.

Test Steps:

```

1. READ NP1, IC =(Issuer_Certificate_Files 1, Issuer_Certificate_Files 2)
4. WHILE (the last read did not indicate 'End Of File') DO {
   TRANSMIT AtomicReadFile-Request,
   'Object Identifier' = IC[1],
   'File Start Position' = (the next unread octet),
   'Requested Octet Count' = MROC
   RECEIVE AtomicReadFile-ACK,
   'End Of File' = TRUE | FALSE,
   'File Start Position' = (the next unread octet)
   'File Data' = (IC[1] File contents of length MROC if 'End Of File' is FALSE
   or of length MROC or less if 'End Of File' is TRUE)
   }
5. WHILE (the last read did not indicate 'End Of File') DO {
   TRANSMIT AtomicReadFile-Request,

```

```
        'Object Identifier' = IC[2],
        'File Start Position' = (the next unread octet),
        'Requested Octet Count' = MROC
    RECEIVE AtomicReadFile-ACK,
        'End Of File' = TRUE | FALSE,
        File Start Position' = (the next unread octet)
        'File Data' = (IC[2] File contents of length MROC if 'End Of File' is FALSE
                      or of length MROC or less if 'End Of File' is TRUE)
    }
6. IF (IC[1] File contents is = the IUT's known current issuer certificate) THEN
    WRITE IC[2], File_Size = 0
    VERIFY IC[2], Modification_Date = (the current local data and time)
ELSE
    WRITE IC[1], File_Size = 0
    VERIFY IC[1], Modification_Date = (the current local data and time)
7. VERIFY NP1, Changes_Pending = TRUE
8. TRANSMIT ReinitializeDevice-Request
    'Reinitialized State of Device' = WARMSTART | ACTIVATE_CHANGES
    'Password' = (any valid password)
9. RECEIVE BACnet-SimpleACK-PDU
10. WAIT Activate Changes Fail Time
11. CHECK (the connection is re-established)
12. VERIFY NP1, Changes_Pending = FALSE
```

**BTL-23.3 cc-2: BACnet/SC Heartbeat and Hub Function Tests [BTLWG-1429, BTLWG-1469]**

**Overview:**

Add Interim test for 135-2020 Addendum cc specifically for Hub and Annex AB changes.

AB.5.3.1 - Node needs to process a Heartbeat request and respond with an ack. (PR 24)

AB.6.3 - New Node Negative test that checks that a node disconnects if a heartbeat-ack is not received. (PR 24)

12.56.Y14 – New test to ensure the SC\_Hub\_Function\_Enable stops being a Hub when configured to FALSE. (PR 24)

AB.5.3.1 – New test to ensure the hub function does a Heartbeat-Request before terminating a connection. (PR 24)

**Changes:**

---

**Checklist Changes**

---

None

---

**Test Plan Changes**

---

[Add new tests to section]

**9.9.2 Is Able to Operate as a Node Without a Local Hub Function**

...

<b>BTL - 12.5.1.1.X1 - Node Heartbeat-Request Execution Test</b>		
	<b>Test Conditionality</b>	If the IUT supports Protocol_Revision 24 or greater, this test must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 12.5.1.2.X1 - Node Heartbeat-Request Initialization Failure Test</b>		
	<b>Test Conditionality</b>	If the IUT supports Protocol_Revision 24 or greater, this test must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

...

**9.9.3 Is Able to Operate as a Hub**

...

<b>BTL - 12.5.2.1.X1 - SC Hub Function Enable Property Test</b>		
	<b>Test Conditionality</b>	If the IUT supports Protocol_Revision 24 or greater, this test must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 12.5.2.2.X1 - Hub Heartbeat-Request Initialization Failure Test</b>		
	<b>Test Conditionality</b>	If the IUT supports Protocol_Revision 24 or greater, this test must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

...

---

## Specified Test Changes

---

[Add 12.5.1.1.X1]

### 12.5.1.1.X1 Node Heartbeat-Request Execution Test

Reference: Addendum cc Clause AB.5.3.1.

Purpose: To verify that a node device accepts and responds to Heartbeat-Requests.

Test Concept: With the TD operating as a hub, the IUT connects to the TD. The TD sends a Heartbeat-Request to the IUT. Verify the IUT responds with a Heartbeat-ACK.

Configuration Requirements: The IUT is configured as a node and connected to the TD.

Test Steps:

1. MAKE(the TD generate a Heartbeat-Request)
2. RECEIVE PORT (TD-IUT hub WebSocket),  
Heartbeat-ACK,  
'Message ID' = M1: any valid value),  
-- 'Originating Virtual Address' absent  
-- 'Destination Virtual Address' absent  
'Destination Options' = (absent or a valid list of options),  
-- 'Data Options' absent

[Add 12.5.1.2.X1]

### 12.5.1.2.X1 Node Heartbeat-Request Initialization Failure Test

Reference: Addendum cc Clause AB.6.3.

Purpose: To verify that a Node will disconnect if a Heartbeat-ACK is not received.

Test Concept: With the IUT connected to the TD as the primary hub, allow the IUT to connect to the TD. OD sends a ReadProperty request to the IUT every HB / 2 seconds. HB is the value of the IUTs SC\_Heartbeat\_Timeout property. Stop sending messages to the IUT. Wait HB plus 10 seconds and verify the IUT sends a Heartbeat-Request, times out waiting for a Heartbeat-ACK and then the IUT sends a Disconnect-Request.

Configuration Requirements: Configure the SC\_Heartbeat\_Timeout property of the TD to be 2 times HB. Place the TD in a mode where it will not respond to Heartbeat-Requests.

Test Steps:

1. REPEAT N = (1..Z) {  
TRANSMIT Encapsulated-NPDU,  
'Message ID' = (M: any valid value),  
'Originating Virtual Address' = (OD's VMAC),  
-- 'Destination Virtual Address' absent  
'Destination Options' (absent or any valid value),  
'Data Options' = ({ X'41'}), -- Secure Path  
'BACnet NPDU' =  
ReadProperty-Request,  
'Object Identifier' = (the IUT's Device object),  
'Property Identifier' = Object\_Name  
RECEIVE Encapsulated-NPDU,  
'Message ID' = M,  
-- 'Originating Virtual Address' absent  
'Destination Virtual Address' = (OD's VMAC),  
'Destination Options' (absent or any valid value),  
'Data Options' = ({ X'41' or a list of valid header options including Secure Path}),

```

        'BACnet NPDU' =
            ReadProperty-ACK,
            'Object Identifier' =      (the IUT's Device object),
            'Property Identifier' =    Object_Name,
            'Property Value' =        (the IUT's device object name)
    WAIT HB / 2
}
-- Since we already waited ½ of HB, only HB / 2 of that interval is now given for the IUT to
-- generate a Heartbeat-Request
2. BEFORE HB / 2 + 10s
    RECEIVE Heartbeat-Request,
        'Message ID' =                (M: any valid value)
        -- 'Originating Virtual Address' absent
        -- 'Destination Virtual Address' absent
        'Destination Options' =        (absent or any valid value),
        -- 'Data Options' absent
3. BEFORE (2 seconds or Vendor specified timeout)
    RECEIVE Disconnect-Request,
        'Message ID' =                (M: any valid value)
        -- 'Originating Virtual Address' absent
        -- 'Destination Virtual Address' absent
        'Destination Options' =        (absent or any valid value),
        -- 'Data Options' absent

```

[Add 12.5.2.1.X1]

### 12.5.2.1.X1 SC\_Hub\_Function\_Enable Property Test

Reference: Addendum cc Clause 12.56.Y14.

Purpose: To ensure the IUTs hub function can be enabled and disabled using the SC\_Hub\_Function\_Enable property.

Test Concept: With the IUTs SC\_Hub\_Function\_Enable property set to TRUE, verify the IUT is operating as a hub. Change the IUTs SC\_Hub\_Function\_Enable property to FALSE and verify the IUT is no longer operating as a hub.

Configuration Requirements: The IUT is configured as the primary hub and the value of the SC\_Hub\_Function\_Enable property to TRUE. ~~The TDs primary hub URI is configured to reference the IUT, and TDs failover hub URI is not configured.~~ The TD is configured as the failover hub. The TDs primary hub URI is configured to reference the IUT, and the IUTs failover hub URI is configured to reference the TD.

Test Steps:

1. MAKE(the TD open a WebSocket to the IUT's hub function)
2. TRANSMIT PORT (TD-IUT hub WebSocket)
  - Connect-Request,
  - 'Message ID' = (M1: any valid value),
  - 'Originating Virtual Address' absent
  - 'Destination Virtual Address' absent
  - 'Destination Options' absent
  - 'Data Options' absent
  - 'VMAC Address' = (TD's VMAC),
  - 'Device UUID' = (TD's UUID),
  - 'Maximum BVLC Length' = (the TD's maximum BVLC accepted length),
  - 'Maximum NPDU Length' = (the TD's maximum NPDU accepted length)
3. RECEIVE PORT (TD-IUT hub WebSocket)
  - Connect-Accept,
  - 'Message ID' = M1,
  - 'Originating Virtual Address' absent
  - 'Destination Virtual Address' absent
  - 'VMAC Address' = (IUT's VMAC),
  - 'Device UUID' = (IUT's UUID),

```

'Maximum BVLC Length' = (the IUT's maximum BVLC accepted length),
'Maximum NPDU Length' = (the IUT's maximum NPDU accepted length)
4. IF (SC_Hub_Function_Enable is writable) THEN
    WRITE SC_Hub_Function_Enable = FALSE
    TRANSMIT ReinitializeDevice-Request
        'Reinitialized State of Device' = WARMSTART | ACTIVATE_CHANGES
        'Password' = (any valid password)
    RECEIVE BACnet-SimpleACK-PDU
ELSE
    MAKE (SC_Hub_Function_Enable = FALSE)
5. TRANSMIT ReinitializeDevice-Request
    'Reinitialized State of Device' = WARMSTART | ACTIVATE_CHANGES
    'Password' = (any valid password)
6. RECEIVE BACnet-SimpleACK-PDU
7. WAIT Activate Changes Fail Time
8. CHECK(that the TD attempts and fails to open a WebSocket to the IUT)
9. IF (SC_Hub_Function_Enable is writable) THEN
    WRITE SC_Hub_Function_Enable = TRUE
ELSE
    MAKE (SC_Hub_Function_Enable = TRUE)
10. TRANSMIT ReinitializeDevice-Request
    'Reinitialized State of Device' = WARMSTART | ACTIVATE_CHANGES
    'Password' = (any valid password)
11. RECEIVE BACnet-SimpleACK-PDU
12. WAIT Activate Changes Fail Time
13. MAKE(the TD open a WebSocket to the IUT's hub function)
9. CHECK(that the IUT opens a WebSocket with the TD)
10. TRANSMIT PORT (TD-IUT hub WebSocket)
    Connect-Request,
    'Message ID' = (M1: any valid value),
    -- 'Originating Virtual Address' absent
    -- 'Destination Virtual Address' absent
    -- 'Destination Options' absent
    -- 'Data Options' absent
    'VMAC Address' = (TD's VMAC),
    'Device UUID' = (TD's UUID),
    'Maximum BVLC Length' = (the TD's maximum BVLC accepted length),
    'Maximum NPDU Length' = (the TD's maximum NPDU accepted length)
11. RECEIVE PORT (TD-IUT hub WebSocket)
    Connect-Accept,
    'Message ID' = M1,
    -- 'Originating Virtual Address' absent
    -- 'Destination Virtual Address' absent
    'VMAC Address' = (IUT's VMAC),
    'Device UUID' = (IUT's UUID),
    'Maximum BVLC Length' = (the IUT's maximum BVLC accepted length),
    'Maximum NPDU Length' = (the IUT's maximum NPDU accepted length)
12. VERIFY (SC_Hub_Function_Enable = FALSE)
13. IF (SC_Hub_Function_Enable is writable) THEN
    WRITE SC_Hub_Function_Enable = TRUE
    TRANSMIT ReinitializeDevice-Request
        'Reinitialized State of Device' = WARMSTART | ACTIVATE_CHANGES
        'Password' = (any valid password)
    RECEIVE BACnet-SimpleACK-PDU
ELSE
    MAKE (SC_Hub_Function_Enable = TRUE)
14. WAIT Activate Changes Fail Time
15. MAKE(the TD open a WebSocket to the IUT's hub function)
16. TRANSMIT PORT (TD-IUT hub WebSocket)
    Connect-Request,

```

```

    'Message ID' = (M1: any valid value),
    -- 'Originating Virtual Address' absent
    -- 'Destination Virtual Address' absent
    -- 'Destination Options' absent
    -- 'Data Options' absent
    'VMAC Address' = (TD's VMAC),
    'Device UUID' = (TD's UUID),
    'Maximum BVLC Length' = (the TD's maximum BVLC accepted length),
    'Maximum NPDU Length' = (the TD's maximum NPDU accepted length)
17. RECEIVE PORT (TD-IUT hub WebSocket)
    Connect-Accept,
    'Message ID' = M1,
    -- 'Originating Virtual Address' absent
    -- 'Destination Virtual Address' absent
    'VMAC Address' = (IUT's VMAC),
    'Device UUID' = (IUT's UUID),
    'Maximum BVLC Length' = (the IUT's maximum BVLC accepted length),
    'Maximum NPDU Length' = (the IUT's maximum NPDU accepted length)

```

[Add 12.5.2.2.X1]

### 12.5.2.2.X1 Hub Heartbeat-Request Initialization Failure Test

Reference: Addendum cc Clause AB.5.3.1.

Purpose: To verify that a Hub initiates a Heartbeat-Request before attempting to terminate a connection.

Test Concept: With the IUT operating as a hub, the TD connects to the IUT. The TD sends a ReadProperty request to the IUT every HB / 2 seconds. HB is the value of the IUTs SC\_Heartbeat\_Timeout property. Stop sending messages to the IUT. Wait HB plus 10 seconds and verify the IUT sends a Heartbeat-Request, times out waiting for a Heartbeat-ACK and then the IUT sends a Disconnect-Request.

Configuration Requirements: Configure the SC\_Heartbeat\_Timeout property of the TD to be 2 times HB. Place the TD in a mode where it will not respond to Heartbeat-Requests.

Test Steps:

```

1. REPEAT N = (1..Z) {
    TRANSMIT Encapsulated-NPDU,
        'Message ID' = (M: any valid value),
        -- 'Originating Virtual Address' absent
        'Destination Virtual Address' = (IUT's VMAC),
        'Destination Options' (absent or any valid value),
        'Data Options' = ({ X'41' }), -- Secure Path
        'BACnet NPDU' =
            ReadProperty-Request,
            'Object Identifier' = (the IUT's Device object),
            'Property Identifier' = Object_Name
    RECEIVE Encapsulated-NPDU,
        'Message ID' = M,
        'Originating Virtual Address' = (IUT's VMAC),
        -- 'Destination Virtual Address' absent
        'Destination Options' (absent or any valid value),
        'Data Options' = ({ X'41' or a list of valid header options including Secure Path}),
        'BACnet NPDU' =
            ReadProperty-ACK,
            'Object Identifier' = (the IUT's Device object),
            'Property Identifier' = Object_Name,
            'Property Value' = (the IUT's device object name)
    WAIT HB / 2
}

```

-- Since we already waited  $\frac{1}{2}$  of HB, only HB / 2 of that interval is now given for the IUT to  
-- generate a Heartbeat-Request

2. BEFORE HB / 2 + 10s

RECEIVE Heartbeat-Request,

'Message ID' = (M: any valid value)

-- 'Originating Virtual Address' absent

-- 'Destination Virtual Address' absent

'Destination Options' = (absent or any valid value),

-- 'Data Options' absent

3. BEFORE (2 seconds or Vendor specified timeout)

RECEIVE Disconnect-Request,

'Message ID' = (M: any valid value)

-- 'Originating Virtual Address' absent

-- 'Destination Virtual Address' absent

'Destination Options' = (absent or any valid value),

-- 'Data Options' absent

**BTL-23.3 cc-3: DISCARD\_CHANGES Command Testing [BTLWG-1493]**

**Overview:**

TP3.56.7 - Support the DISCARD\_CHANGES Command does not test for a B/SC certificate as of PR24.

**Changes:**

---

**Checklist Changes**

---

None

---

**Test Plan Changes**

---

**3.56-6.7 Supports the DISCARD\_CHANGES Command**

The IUT supports the DISCARD\_CHANGES command in Network Port objects.

<b>135.1-2023 - 7.3.2.46.3.2 - DISCARD_CHANGES Command Test</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 7.3.2.46.3.2.X.5 - DISCARD_CHANGES Command With File Object References Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT claims Protocol_Revision >= 24 and supports BACnet/Secure Connect.
	<b>Test Directives</b>	Repeat this test by writing to the File object referenced in the Operational_Certificate_File property and each of the File objects referenced in the Issuer_Certificate_Files property of each Network Port object where the Network_Type = SECURE_CONNECT.
	<b>Testing Hints</b>	

---

**Specified Test Changes**

---

[Add 7.3.2.46.3.X.5]

**7.3.2.46.3.X.5 DISCARD\_CHANGES Command With File Object References Test**

Reason for Change: New test per Addendum 135-2020cc Clauses 12.56.Y24 and 12.56.Y25.

Purpose: To verify that the Network Port object and linked File objects discard pending changes when the Command DISCARD\_CHANGES is received.

Test Concept: Write the File object linked to the Network Port object, verify the write was successful, write DISCARD\_CHANGES to the Command property of the Network Port object, and verify that the File object and Network Port object properties revert to their previous values. Repeat the test writing the File object, File\_Size to zero.

Configuration Requirements: Execute this test on a Network Port object, NP1, with Network\_Type = SECURE\_CONNECT and supports the DISCARD\_CHANGES command. F1 is the File object referenced by a property of NP1. When performing the AtomicWriteFile service, a Maximum Write Data Length (MWDL) shall be calculated before starting the test. MWDL shall be 21 octets less than the minimum of the TD's maximum transmittable APDU length and the IUT's Max\_APDU\_Length\_Accepted.

**Test Steps:**

- write to the File object
- 1. VERIFY NP1, Changes\_Pending = FALSE

2. VERIFY NP1, Command = IDLE
3. READ FS1 = F1, File\_size
4. READ MD1 = F1, Modification\_Date
5. REPEAT Z = (0 through the file size, FS2, in increments of MWDL) DO {  
    TRANSMIT AtomicWriteFile-Request  
    'File Identifier' = F1  
    'File Start Position' = Z  
    'Record Data' = (any valid file content, the number of octets being the lesser of  
        (file size - Z) and MWDL)  
    RECEIVE AtomicWriteFile-ACK  
    'File Start Position' = Z  
}
6. VERIFY NP1, Changes\_Pending = TRUE
7. VERIFY F1, File\_Size = FS2
8. VERIFY F1, Modification\_Date = (the current local date and time)
  
- discard changes
9. WRITE NP1, Command = DISCARD\_CHANGES
10. WAIT Activate Changes Fail Time
  
- verify the Network Port object was successfully reverted
11. VERIFY NP1, Changes\_Pending = FALSE
12. VERIFY NP1, Command = IDLE
  
- verify the File object was successfully reverted
13. VERIFY F1, File\_Size = FS1
14. VERIFY F1, Modification\_Date = (MD1 or the current local date and time)
15. VERIFY F1, Archive = FALSE
  
- write File\_Size = 0
16. WRITE F1, File\_Size = 0
17. VERIFY NP1, Changes\_Pending = TRUE
  
- verify the File object was successfully written
18. VERIFY F1, File\_Size = 0
19. VERIFY F1, Modification\_Date = (the current local date and time)
  
- discard changes
20. WRITE NP1, Command = DISCARD\_CHANGES
21. WAIT Activate Changes Fail Time
  
- verify the Network Port object was successfully reverted
22. VERIFY NP1, Changes\_Pending = FALSE
23. VERIFY NP1, Command = IDLE
  
- verify the File object was successfully reverted
24. VERIFY F1, File\_Size = FS1
25. VERIFY F1, Modification\_Date = (MD1 or the current local date and time)
26. VERIFY F1, Archive = FALSE

## BTL-23.3 cc-4: Fix Direct Connect Establishment Test [BTLWG-1517]

### Overview:

It was noted the current version of test 12.5.3.3.1.1 (Direct Connect Establishment Test) requires a Direct Connection (DC) in Step 1 which occurs prior to the DC URI being obtained by the IUT. This needs to be fixed.

### Changes:

---

## Checklist Changes

---

None

---

## Test Plan Changes

---

[In BTL Test Plan, modify all references of test 12.5.3.3.1.1 from 135.1-2023 to BTL]

---

## Specified Test Changes

---

[Move tests 12.5.3.3.1.1 into BTL Specified Tests, and modify]

### 12.5.3.3.1.1 Direct Connect Establishment Test

*Reason for change: The original test incorrectly expected the creation of a Direct Connection before the peer's direct-connect URI is discovered.*

Purpose: To verify that the IUT is able to correctly establish a direct connection with a non-hub peer BACnet/SC node.

Test Concept: With IUT connected to the network, make the IUT establish a direct connection to another node on the network. Verify that the direct connection is correctly established.

Configuration Requirements: The IUT is configured to support establishing direct connections. The IUT is connected to the primary hub. D3 is configured to support accepting direct connections. The IUT is configured to use dynamic discovery of WebSocket-URIs if supported, otherwise the WebSocket-URI for D3 is configured into the IUT.

### Test Steps:

1. MAKE (an action that will cause the IUT to initiate establish a direct connection to D3)
2. IF (the IUT supports discovering direct connection URIs) {
  - RECEIVE PORT (IUT-TD hub WebSocket)
    - Address-Resolution,
    - 'Message ID' = (M1: any valid value),
    - 'Originating Virtual Address' absent
    - 'Destination Virtual Address' = D3,
    - 'Destination Options' = (absent or a list of valid options),
    - 'Data Options' absent
  - TRANSMIT PORT (IUT-TD hub WebSocket)
    - Address-Resolution-ACK,
    - 'Message ID' = M1,
    - 'Originating Virtual Address' = D3
    - 'Destination Virtual Address' absent
    - 'Destination Options' absent
    - 'Data Options' absent
    - 'Payload'
    - 'WebSocket-URIs' W: a WebSocket URI which D3 can be reached at)

3. CHECK(that the IUT opens a WebSocket to D3's WebSocket-URI)
4. RECEIVE PORT (IUT-D3 direct connect WebSocket),
  - Connect-Request,
  - 'Message ID' = (M2: any valid value),
  - 'Originating Virtual Address' absent
  - 'Destination Virtual Address' absent
  - 'Destination Options' (absent or any valid value),
  - 'Data Options' absent
  - 'VMAC Address' = (IUT's VMAC),
  - 'Device UUID' = (IUT's UUID),
  - 'Maximum BVLC Length' = (the IUT's maximum BVLC accepted length),
  - 'Maximum NPDU Length' = (the IUT's maximum NPDU accepted length)
5. TRANSMIT Connect-Accept,
  - 'Message ID' = M2,
  - 'Originating Virtual Address' absent
  - 'Destination Virtual Address' absent
  - 'Destination Options' (absent or any valid value),
  - 'Data Options' absent
  - 'VMAC Address' = (D3's VMAC),
  - 'Device UUID' = (D3's UUID),
  - 'Maximum BVLC Length' = (the D3's maximum BVLC accepted length),
  - 'Maximum NPDU Length' = (the D3's maximum NPDU accepted length)

**BTL-23.3 cc-5: New Tests for Network Port Object and B/SC [BTLWG-1477]**

**Overview:**

Add tests for Network Port and B/SC as per 135-2020cc

**Changes:**

**Checklist Changes**

[Update the two BIBBs in the checklist to remove "Contact BTL for interim tests for this BIBB" and add Protocol\_Revision 24 or higher must be claimed.]

**10 Network Management**

<b>Network Management - Communications Configuration - A</b>		
	R <sup>1</sup>	Base Requirements
		<sup>1</sup> Protocol Revision 24 or higher must be claimed
<b>Network Management - Secure Connect Certificate Management - A</b>		
	R <sup>1</sup>	Base Requirements
		<sup>1</sup> Protocol Revision 24 or higher must be claimed

**Test Plan Changes**

[Add new section 10.X NM-CC-A and 10.Y NM-SCCM-A]

**10.X Network Management - Communications Configuration - A**

**10.X.1 Base Requirements**

Base requirements must be met by any IUT claiming conformance to this BIBB.

<b>Verify Checklist</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Verify that the IUT claims support for DS-WP-A.
	<b>Testing Hints</b>	
<b>Verify Checklist</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Verify that the IUT claims support for DS-RP-A.
	<b>Testing Hints</b>	
<b>Verify Epics</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Verify that the IUT claims support for Initiate Atomic Write File
	<b>Testing Hints</b>	
<b>Verify Epics</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Verify that the IUT claims support for Initiate Atomic Read File
	<b>Testing Hints</b>	
<b>Verify Checklist</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Verify that the IUT claims support for DM-RD-A
	<b>Testing Hints</b>	
<b>135.1-2023 - 8.18.3 - Reading and Presenting Properties</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	Repeat the test for all standard properties of the Network Port object type.
	<b>Testing Hints</b>	
<b>135.1-2023 - 8.22.4 - Accepting Input and Modifying Properties</b>		

<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Repeat the test for all standard properties of the Network port object type except any property defined by the standard as read-only, or to which access is otherwise restricted by the standard. Repeat the test for a variety of values that cover the range of values required by the “Minimum Writable Value Ranges” table in the DS-M-A BIBB definition.
<b>Testing Hints</b>	

## 10.Y Network Management - Secure Connect Certificate Management - A

### 10.Y.1 Base Requirements

Base requirements must be met by any IUT claiming conformance to this BIBB.

<b>Verify Checklist</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Verify that the IUT claims support for DS-WP-A.
<b>Testing Hints</b>	
<b>Verify Checklist</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Verify that the IUT claims support for DS-RP-A.
<b>Testing Hints</b>	
<b>Verify Epics</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Verify that the IUT claims support for Initiate Atomic Write File
<b>Testing Hints</b>	
<b>Verify Epics</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Verify that the IUT claims support for Initiate Atomic Read File
<b>Testing Hints</b>	
<b>Verify Checklist</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	Verify that the IUT claims support for DM-RD-A
<b>Testing Hints</b>	
<b>BTL - 13.Y.1 - Extending Operational Certificates' Validity Period</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	
<b>BTL - 13.Y.2 - Extending Issuer Certificate Validity Period</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	
<b>BTL - 13.Y.3 - Banning One or More Nodes from the Network</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	

---

## Specified Test Changes

---

[Add new tests to BTL Specified Tests]

### 13.Y.1 Extending Operational Certificates' Validity Period

Reason for Change: There is no test for this functionality.

Purpose: To verify the IUT replaces the BACnet/SC Operational certificates in all nodes and hubs in a BACnet/SC network.

Test Concept: The IUT is made to initiate an update of the operational certificates in all nodes in a BACnet/SC network. This test verifies the successful replacement of the operational certificates by ensuring each node is able to reconnect to the BACnet/SC network.

Configuration Requirements: This test assumes each device on the BACnet/SC network is active and is at Protocol\_Revision 24 or greater. The signing CA must be an external entity.

Notes To Tester: Individual steps can be performed on one or more nodes. For example, the IUT is allowed to upload all the CSRs before delivering them to the signing CA.

Test Steps:

1. MAKE (IUT initiate the process to replace the Operational Certificates in every node in a BACnet/SC network)
2. REPEAT X = (all nodes in the network with the hub node done last) DO
3.     CHECK (that the IUT attempts to write to the GENERATE\_CSR\_FILE command)
4.     IF (the TD return an BACnet-SimpleACK) THEN
5.         CHECK (that the IUT wait for Command=IDLE)
6.     CHECK (that the IUT uploads the File object that contains the CSR)
7.     CHECK (that the IUT makes the uploaded CSR available to the signing CA and accepts the new operational certificate based on the uploaded CSR)
8.     CHECK (that the IUT downloads the new operational certificate to the File object that represents the operational certificate)
9.     CHECK (that the IUT sends a ReinitializeDevice (ACTIVATE\_CHANGES or WARMSTART, <password>) message to activate the changes to the Network Port object)
10.    WAIT (Activate Changes Fail Time specified for X)
11.    CHECK (the connection to X is re-established)
- }

### 13.Y.2 Extending Issuer Certificates' Validity Period

Reason for Change: There is no test for this functionality.

Purpose: To verify the IUT replaces the BACnet/SC Issuer certificates in all nodes and hubs in a BACnet/SC network.

Test Concept: The IUT is made to initiate a download of a new issuer certificates, replace the operational certificates and remove the old issuer certificates of a BACnet/SC network. This test verifies the successful replacement of the issuer certificates by ensuring each node is able to reconnect to the BACnet/SC network.

Configuration Requirements: This test assumes each device on the BACnet/SC network is active and is at Protocol\_Revision 24 or greater. The signing CA must be an external entity.

Notes To Tester: Individual steps can be performed on one or more nodes.

Test Steps:

1. MAKE (IUT initiate the Process to replace the issuer Certificates in every node in a BACnet/SC network)  
----- Add a new issuer certificate to every node
2. REPEAT X = (all nodes in the network with the hub node done last) DO {
3.     CHECK (that the IUT determines the File object that represents the unused issuer certificate)
4.     CHECK (that the IUT downloads the new issuer certificate to the File object that represents the unused issuer certificate)
5.     CHECK (that the IUT sends a ReinitializeDevice (ACTIVATE\_CHANGES or WARMSTART, <password>) message to activate the changes to the Network Port object)
6.     WAIT (Activate Changes Fail Time specified for X)
7.     CHECK (the connection to X is re-established)

}

----- Replace the operational certificate in every node in the network

8. REPEAT X = (all nodes in the network with the hub node done last) DO {
  9. CHECK (that the IUT attempts to writes the GENERATE\_CSR\_FILE command)
  10. IF (the TD returns a BACnet-SimpleACK) THEN
  11. CHECK (that the IUT waits for Command=IDLE)
  12. CHECK (that the IUT uploads the File object that contains the CSR)
  13. CHECK (that the IUT makes the uploaded CSR available to the signing CA and accepts the new operational certificate based on the uploaded CSR)
  14. CHECK (that the IUT downloads the new operational certificate to the File object that represents the operational certificate)
  15. CHECK (that the IUT sends a ReinitializeDevice (ACTIVATE\_CHANGES or WARMSTART, <password>) message to activate the changes to the Network Port object)
  16. WAIT (Activate Changes Fail Time specified for X)
  17. CHECK (the connection to X is re-established)
- }

----- remove the old issuer certificate from every node in the network

18. REPEAT X = (all nodes in the network with the hub node done last) DO {
  19. CHECK (that the IUT determines the File object that represents the unused issuer certificate)
  20. CHECK (that the IUT deletes the content of the issuer certificate File object that represents the unused issuer certificate)
  21. CHECK (that the IUT sends a ReinitializeDevice (ACTIVATE\_CHANGES or WARMSTART, <password>) message to activate the changes to the Network Port object)
  22. WAIT (Activate Changes Fail Time specified for X)
  23. CHECK (the connection to X is re-established)
- }

### 13.Y.3 Banning One or More Nodes from the Network

Reason for Change: There is no test for this functionality.

Purpose: This test verifies that the IUT can remove a node from a BACnet/SC network.

Test Concept: The IUT is made to initiate a download of new issuer certificates, replace the operational certificates, and remove the old issuer certificates of a BACnet/SC network while skipping the nodes to be removed. This test verifies that the IUT can remove certain nodes from a BACnet/SC network by ensuring that only the chosen nodes can reconnect to the BACnet/SC network and that the nodes designated for removal are no longer part of the network.

Configuration Requirements: This test assumes each device on the BACnet/SC network is active and is at Protocol\_Revision 24 or greater. The signing CA must be an external entity.

Notes To Tester: Individual steps can be performed on one or more nodes.

Test Steps:

1. MAKE (the IUT to initiate the process to ban one or more nodes from the Network)
  - Add a new issuer certificate to each node in the network skipping the nodes to be removed
  2. REPEAT X = (all nodes in the network, except the nodes to be removed, with the hub node being done last) DO {
  3. CHECK (that the IUT determines the File object that represents the unused issuer certificate)
  4. CHECK (that the IUT downloads the new issuer certificate to the File object that represents the unused issuer certificate)
  5. CHECK (that the IUT sends a ReinitializeDevice (ACTIVATE\_CHANGES or WARMSTART, <password>) message to activate the changes to the Network Port object)
  6. WAIT (Activate Changes Fail Time specified for X)
  7. CHECK (the connection to X is re-established)
- }

----- Replace the operational certificate in every node in the network skipping the nodes to be removed.

8. REPEAT X = all nodes in the network, except the nodes to be removed, with the hub node being done last) DO {

9. CHECK (that the IUT attempts to write to the GENERATE\_CSR\_FILE command)
  10. IF (the TD returns a BACnet-SimpleACK) THEN
  11. CHECK (that the IUT waits for Command=IDLE)
  12. CHECK (that the IUT uploads the File object that contains the CSR)
  13. CHECK (that the IUT makes the uploaded CSR available to the signing CA and accepts the new operational certificate based on the uploaded CSR)
  14. CHECK (that the IUT downloads the new operational certificate to the File object that represents the operational certificate)
  15. CHECK (that the IUT sends a ReinitializeDevice (ACTIVATE\_CHANGES or WARMSTART, <password>) message to activate the changes to the Network Port object)
  16. WAIT (Activate Changes Fail Time specified for X)
  17. CHECK (the connection to X is re-established)
- }

----- Remove issuer certificate from every node in the network skipping the removed nodes

- 18.. REPEAT X = all nodes in the network, except the nodes to be removed, with the hub node being done last) DO {
  19. CHECK (that the IUT determines the File object that represents the unused issuer certificate)
  20. CHECK (that the IUT deletes the content of the issuer certificate File object that represents the unused issuer certificate)
  21. CHECK (that the IUT sends a ReinitializeDevice (ACTIVATE\_CHANGES or WARMSTART, <password>) message to activate the changes to the Network Port object)
  22. WAIT (Activate Changes Fail Time specified for X)
  23. CHECK (the connection to X is re-established)
- }

**BTL-23.3 cc-6: DISCARD\_CHANGES Command with Non-Writable NPOs [BTLWG-1571]**

**Overview:**

A B/SC device at Protocol Revision 24 can support DISCARD\_CHANGES and not support any writable properties in the B/SC Network Port object. DISCARD\_CHANGES can be used to revert the File objects referenced by the Operational\_Certificate\_File or Issuer\_Certificate\_Files properties.

**Proposed Changes:**

---

**Checklist Changes**

---

None

---

**Test Plan Changes**

---

[Per cc-1 above, this section is now 3.56.7]

**3.56.67 Supports the DISCARD\_CHANGES Command**

The IUT supports the DISCARD\_CHANGES command in Network Port objects.

<b>135.1-2023 - 7.3.2.46.3.2 - DISCARD_CHANGES Command Test</b>	
<b>Test Conditionality</b>	<del>Must be executed if the IUT supports the DISCARD_CHANGES command.</del> <i>If the IUT claims 'Supports Procedure to Replace BACnet/SC Certificates' and does not support a write to any writable property of the Network Port Object that causes the Changes_Pending property to be TRUE, this test shall be skipped, otherwise this test shall be executed.</i>
<b>Test Directives</b>	<i>Repeat this test on each Network Port object that supports DISCARD_CHANGES.</i>
<b>Testing Hints</b>	
<b>BTL - 7.3.2.46.3.2.X.5 - DISCARD_CHANGES Command With File Object References Test</b>	
<b>Test Conditionality</b>	<i>Must be executed if the IUT claims Protocol_Revision &gt;= 24 and supports BACnet/Secure Connect.</i>
<b>Test Directives</b>	<i>Repeat this test by writing to the File object referenced in the Operational_Certificate_File property and each of the File objects referenced in the Issuer_Certificate_Files property of each Network Port object where the Network Type = SECURE CONNECT.</i>
<b>Testing Hints</b>	

---

**Specified Test Changes**

---

None