



**BACnet<sup>®</sup> TESTING LABORATORIES  
ADDENDA**

**Addendum ai to  
BTL Test Package 16.1**

**Revision 5  
Revised April 24, 2020**

Approved by the BTL Working Group on February 13, 2020.  
Approved by the BTL Working Group Voting Members on April 6, 2020.  
Republished on April 24, 2020.

**[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-16.1ai-1: Network Port Object Testing - BTLWG-79.....	2
BTL-16.1ai-2: DM-RD-B Negative Testing - BTLWG-92 .....	31

In the following document, language to be added to existing clauses within the BTL Test Package 16.1 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-16.1ai-1: Network Port Object Testing - BTLWG-79**

**Overview:**

Develop full testing of the Network Port object.

**Changes:**

**BTL Checklist Changes**

[In BTL Checklist, replace Network Port Object section]

Support	Listing	Option
<b>Network Port Object</b>		
	R <sup>1</sup>	Base Requirements
	C <sup>2</sup>	Supports writable Network_Number property
	S	Supports writable Out_Of_Service property
	O	Contains an object with Reliability_Evaluation_Inhibit property
	O	Supports hierarchical Network Port objects
	O	Supports the Command property
	O <sup>3</sup>	Supports the DISCARD_CHANGES command
	O <sup>3</sup>	Supports the RENEW_FD_REGISTRATION command
	O <sup>3</sup>	Supports the RESTART_SLAVE_DISCOVERY command
	O <sup>3</sup>	Supports the RENEW_DHCP command
	O <sup>3</sup>	Supports the RESTART_AUTONEGOTIATION command
	O <sup>3</sup>	Supports the DISCONNECT command
	O <sup>3</sup>	Supports the RESTART_PORT 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 if the Command property is supported.		

[In BTL Checklist, add into each Data Link Layer section]

Support	Listing	Option
	O	Supports configuration through Network Port object

[In BTL Checklist, add into each BACnet/IP Data Link Layer section]

Support	Listing	Option
	O	Supports the Network Port object and DHCP

[In BTL Checklist, add into BACnet/IPv6 Data Link Layer section]

Support	Listing	Option
	O	Supports DHCP

## BTL Test Plan Changes

[In BTL Test Plan, replace Network Port section]

### 3.56 Network Port Object

#### 3.56.1 Base Requirements

Base requirements must be met by any IUT that can contain Network Port objects.

<b>BTL - 7.3.2.X62.1.3 - Network Port Non-Volatility Properties Test</b>		
	<b>Test Conditionality</b>	Must be executed if any writable properties are supported for which the values are required for proper operation of the network.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 7.3.2.X62.1.4 - Network Port Configuration Conflict Test</b>		
	<b>Test Conditionality</b>	If the IUT supports WriteProperty, the test must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 9.18.1.X5 - ReadProperty of the Network Port Object using the Unknown Instance</b>		
	<b>Test Conditionality</b>	Must be executed
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 9.20.1.X3 - ReadPropertyMultiple of the Network Port Object using the Unknown Instance</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 7.3.2.X62.5 - APDU Length Test</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

#### 3.56.2 Supports writable Network\_Number property

The Network\_Number property in Network Port objects contained in the IUT is writable.

<b>BTL - 7.3.2.X62.2 - Network-Number-Is Updates Network Number Quality Test</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	

<b>Testing Hints</b>	
----------------------	--

### 3.56.3 Supports writable Out\_Of\_Service property

The Out\_Of\_Service property in Network Port objects contained in the IUT is either writable or can be modified by any other means.

<b>BTL - 7.3.1.1.2 - Out_Of_Service, Status_Flags, and Reliability test for Objects without Present Value</b>		
	<b>Test Conditionality</b>	If this property is writable, this test must be executed.
	<b>Test Directives</b>	This test shall be applied to a Network Port object.
	<b>Testing Hints</b>	

### 3.56.4 Contains an object with Reliability\_Evaluation\_Inhibit Property

The IUT contains, or can be made to contain, a Reliability\_Evaluation\_Inhibit property that is configurable to a value of TRUE.

<b>BTL - 7.3.1.X8.1 - Reliability_Evaluation_Inhibit Test</b>		
	<b>Test Conditionality</b>	If no object exists in the IUT for which fault conditions can be generated then this test shall be skipped.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 7.3.1.X8.2 - Reliability_Evaluation_Inhibit Summarization Test</b>		
	<b>Test Conditionality</b>	If no object exists in the IUT for which fault conditions can be generated then this test shall be skipped.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

### 3.56.5 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>BTL - 7.3.2.X62.4.1 - 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>BTL - 7.3.2.X62.4.2 - Properties in Referenced Network Port Reflected in Top Network Port Object</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>	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>BTL - 7.3.2.X62.4.3 - Changes Reflected in Top Network Port Object</b>		
	<b>Test Conditionality</b>	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>BTL - 7.3.2.X62.4.4 - Changes Reflected in Lower Network Port Objects</b>		

	<b>Test Conditionality</b>	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.

### 3.56.6 Supports the Command property

The IUT support the Command property in Network Port objects.

<b>BTL - 7.3.2.X62.3.1 - IDLE Command Rejected</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 7.3.2.X62.3.9 - No Commands if Changes Pending Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports a writable property which uses the pending changes mechanism.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

### 3.56.7 Supports the DISCARD\_CHANGES Command

<b>BTL - 7.3.2.X62.3.2 - DISCARD_CHANGES Command Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports the DISCARD_CHANGES command.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

### 3.56.8 Supports the RENEW\_FD\_REGISTRATION Command

<b>BTL - 7.3.2.X62.3.3.1 - RENEW_FD_REGISTRATION Command Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports the RENEW_FD_REGISTRATION command and BACnet/IP or BACnet/IPv6.
	<b>Test Directives</b>	Repeat for BACnet/IP and BACnet/IPv6, if supported.
	<b>Testing Hints</b>	
<b>BTL - 7.3.2.X62.3.3.2 - RENEW_FD_REGISTRATION Command Failure Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports a Network Port object for which RENEW_FD_REGISTRATION is not applicable or not supported.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

### 3.56.9 Supports the RESTART\_SLAVE\_DISCOVERY Command

<b>BTL - 7.3.2.X62.3.4.1 - RESTART_SLAVE_DISCOVERY Command Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports the RESTART_SLAVE_DISCOVERY command.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 7.3.2.X62.3.4.2 - RESTART_SLAVE_DISCOVERY Command Failure Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports a Network Port object for which RESTART_SLAVE_DISCOVERY is not applicable or not supported.

	<b>Test Directives</b>	
	<b>Testing Hints</b>	

### 3.56.10 Supports the RENEW\_DHCP Command

<b>BTL - 7.3.2.X62.3.5.1 - RENEW_DHCP Command Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports the RENEW_DHCP command.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 7.3.2.X62.3.5.2 - RENEW_DHCP Command Failure Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports a Network Port object for which RENEW_DHCP is not applicable or not supported.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

### 3.56.11 Supports the RESTART\_AUTONEGOTIATION Command

<b>BTL - 7.3.2.X62.3.6.1 - RESTART_AUTONEGOTIATION Command Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports the RESTART_AUTONEGOTIATION command.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 7.3.2.X62.3.6.2 - RESTART_AUTONEGOTIATION Command Failure Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports a Network Port object for which RESTART_AUTONEGOTIATION is not applicable or not supported.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

### 3.56.12 Supports the DISCONNECT Command

<b>BTL - 7.3.2.X62.3.7.1 - DISCONNECT Command Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports the DISCONNECT command.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 7.3.2.X62.3.7.2 - DISCONNECT Command Failure Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports a Network Port object for which DISCONNECT is not applicable or not supported.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

### 3.56.13 Supports the RESTART\_PORT Command

<b>BTL - 7.3.2.X62.3.8.1 - RESTART_PORT Command Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports the RESTART_PORT command.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 7.3.2.X62.3.8.2 - RESTART_PORT Command Failure Test</b>		
	<b>Test Conditionality</b>	Must be executed if the IUT supports a Network Port object for which RESTART_PORT is not supported.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

### 3.56.14 Supports the Routing\_Table Property

<b>BTL - 7.3.2.X62.6 - Routing Table Test</b>	
<b>Test Conditionality</b>	If the IUT only supports 1 entry in its routing table, then this test shall be skipped.
<b>Test Directives</b>	
<b>Testing Hints</b>	



## 9.X Data Link Layer - (each)

[In BTL Test Plan, append entry to each Data Link Layer Base Requirements section]

### 9.X.1 Base Requirements

<b>BTL - 7.3.2.X62.1.2 - Verify Network Configuration 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.
<b>Testing Hints</b>	

[In BTL Test Plan, add new section into each Data Link Layer]

### 9.X.Y Supports Configuration Through Network Port Object

<b>BTL - 7.3.2.X62.1.1 - 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.
<b>Testing Hints</b>	

## 9.3 & 9.4 BACnet/IP ...

[In BTL Test Plan, add a new section into each BACnet/IP Data Link Layer section]

### 9.X.Y Supports Network Port Objects and DHCP

<b>BTL - 7.3.2.X62.7.1 - Basic IP DHCP Test</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	

## 9.9 BACnet/IPv6

[In BTL Test Plan, add a new section into each BACnet/IPv6 Data Link Layer section]

### 9.9.Y Supports DHCP

<b>BTL - 7.3.2.X62.7.2 - Basic IPv6 DHCP Test</b>	
<b>Test Conditionality</b>	Must be executed.
<b>Test Directives</b>	
<b>Testing Hints</b>	

## 9.4 BACnet/IP - Annex J - BBMD

[In BTL Test Plan, modify 9.4.2]

### 9.4.2 Supports a BDT with at Least Four Entries

The IUT acts as, or can be made to act as, a BBMD with a BDT that supports at least four entries in its BDT.

This functionality is required of all devices that claim support for an Annex J BACnet/IP BBMD.

<b>135.1-2013 - 14.3.1 - Execute Write-Broadcast-Distribution-Table (Table Growth)</b>		
	<b>Test Conditionality</b>	If the IUT does not support the Write-Broadcast-Distribution-Table message, this test may be skipped. <i>If the IUT claims Protocol_Revision &gt;= 17, this test shall be skipped.</i>
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>135.1-2013 - 14.3.2 - Execute Write-Broadcast-Distribution-Table (Table Shrinkage)</b>		
	<b>Test Conditionality</b>	If the IUT does not support the Write-Broadcast-Distribution-Table message, this test may be skipped. <i>If the IUT claims Protocol_Revision &gt;= 17, this test shall be skipped.</i>
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>BTL - 14.3.3 - Verify Broadcast Distribution Table Created from the Configuration Saved During the Previous Session</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
<b>135.1-2013 - 14.5.2.2 - Original-Broadcast-NPDU Which Shall Be Forwarded (Two-hop Distribution)</b>		
	<b>Test Conditionality</b>	Must be executed.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

[In BTL Test Plan, add test to end of 9.4.1 Base Requirements for BACnet/IP - Annex J - BBMD]

<b>BTL - 14.3.X1 - Write-BDT service is required to return Write-BDT-NAK</b>		
	<b>Test Conditionality</b>	Must be executed in all devices claiming Protocol Revision >= 17.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	

## BTL Specified Tests Changes

[In BTL Specified Tests, modify Clause 4.5.9]

[Add the following into the list of timers]

Auto Negotiation Fail Time: \_J

[In BTL Specified Tests, add Clause 6.3.X]

### 6.3.X Auto Negotiation Fail Time

The **Auto Negotiation Fail Time** is the elapsed time, in seconds, between when auto negotiation is requested and when a test is considered to have failed because the IUT has not completed auto negotiation of link speed.

[In BTL Specified Tests, modify 7.2.2]

### 7.2.2 Write Support Test Procedure

Reason for Change: 'Notes to Tester' is missing from the version in 135.1-2013. Added in special handling for properties in the Network Port object. Moved content from the Purpose into Test Concept as appropriate.

Purpose: To verify that all writable properties of all objects can be written to using BACnet WriteProperty and WritePropertyMultiple services. ~~The test is performed once using WriteProperty and once using WritePropertyMultiple. When writing to array properties, the whole array shall be written without using an array index, where possible.~~

*Test Concept: Each writable property property is written multiple times verifying the writable range. After each write, the value is verified to have been updated in the property. The test is performed once using WriteProperty and once using WritePropertyMultiple. When writing to array properties, the whole array shall be written without using an array index, where possible.*

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

Test Steps:

1. REPEAT X = (all objects in the IUT's database, *except Network Port objects*) DO {  
    REPEAT Y = (all writable properties in object X) DO {  
        REPEAT Z = (all values meeting the functional range requirements of 7.2.1, and any additional restrictions placed on the allowable property values by the vendor) DO {  
            WRITE (X), Y = Z,  
            VERIFY (X), Y = Z  
        }  
    }  
}

*Notes to Tester: An internal process may set the Present\_Value of some properties back to the default value after a successful write, as in the case of a momentary pushbutton, or the Record\_Count property. For properties that exhibit this type of behavior, skip the VERIFY step.*

*Notes to Tester: When a property is currently not writable, the IUT shall return an Error-PDU with 'Error Class' = PROPERTY and 'Error Code' = WRITE\_ACCESS\_DENIED.*

[In BTL Specified Tests, add clause 7.3.2.X62]

### 7.3.2.X62 Network Port Object Tests

Configuration Requirements: In addition to the requirements listed for each test, the Network Port object which is being tested shall be configured and operating and have no changes pending, unless required by the specific test's specification.

#### 7.3.2.X62.1 Network Port Configuration Tests

These tests verify that Network Port objects reflect the configuration in use, and for writable ones, change the network configuration.

##### 7.3.2.X62.1.1 Configure Network Through Network Port Object Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: This test verifies that Network Port properties control aspects of the network configuration as expected.

Test Concept: Given the complexity of the Network Port object, and the impact changes to the Network Port has on the test network, this test is provided to allow testing of the Network Port functionality as the network is reconfigured for other tests. The Network Port object is modified to meet the conditions of the new test network setup. The changes are activated, the TD is reconfigured to match, and communication with the IUT is re-verified. The configuration of the network is expected to be tested in more detail as the other datalink tests are applied.

Test Configuration: The test network is configured such that the TD and IUT can communicate, but the configuration does not match the target network configuration. P1 through PN are Network Port properties that need to be written in order to transition the network from the current setup to the target network setup. This set of properties shall be selected from the set of the properties that are writable in the IUT.

Test Steps:

1. REPEAT P = P1 ... PN {
  - WRITE P = (NV: the value required for the target network setup)
  - VERIFY P = NV
2. VERIFY Changes\_Pending = TRUE
3. REPEAT P = P1 ... PN {
  - CHECK (the new value for P is not in use by the network port, unless the new value is the same as the old value)
4. TRANSMIT ReinitializeDevice-Request
  - 'Reinitialized State of Device' = WARMSTART | ACTIVATE\_CHANGES
  - 'Password' = (any valid password)
5. RECEIVE BACnet-SimpleACK-PDU
6. MAKE(change the TD network setup and the network setup of all other devices on the network to match the target network setup)
7. WAIT Activate Changes Fail Time
8. VERIFY Changes\_Pending = FALSE

### 7.3.2.X62.1.2 Verify Network Configuration Through Network Port Object Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: This test verifies that Network Port properties correctly reflect aspects of the network configuration as expected.

Test Concept: Given the complexity of the Network Port object, and the impact changes to the Network Port has on the test network, this test is provided to allow testing of the Network Port functionality as the network is reconfigured for other tests. The IUT's network configuration is modified to meet the conditions of the new test network setup. The TD is reconfigured to match, and communication with the IUT is re-verified. The Network Port object is then checked to ensure it reflects the new network setup.

Test Steps:

1. MAKE(configure the network, including reconfiguring the TD, IUT, and other devices on the network)
2. CHECK(that the value of each of the present Network Port properties which applies to the associated data link reflects the current network setup)

### 7.3.2.X62.1.3 Network Port Non-Volatility Properties Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: This test verifies that after Network Port properties are changed, and activated, the revised value is maintained through a power failure and device restart.

Test Concept: Write one or more properties, P1 ... PN, of a Network Port object which are required for proper operation of the network port. If any of the properties utilize the pending changes functionality, activate the changes. Restart the IUT device by temporarily removing power. When the device has resumed operation after that restart, verify that the new values for the properties were maintained across the reset and are in use by the port.

Test Steps:

1. REPEAT P = P1 ... PN {
  - WRITE P = (a new value different from the property's current value)

2. IF any of the properties utilize the pending change functionality THEN
  - VERIFY Changes\_Pending = TRUE
  - TRANSMIT ReinitializeDevice-Request
    - 'Reinitialized State of Device' = WARMSTART | ACTIVATE\_CHANGES
    - 'Password' = (any valid password)
  - RECEIVE BACnet-SimpleACK-PDU
  - MAKE(reconfigure the TD and other devices on the network to the new network settings)
  - WAIT Activate Changes Fail Time
- ELSE
  - VERIFY Changes\_Pending = FALSE
3. REPEAT P = P1 ... PN {
  - VERIFY P = (the new value for the property)
4. MAKE (the IUT power cycle to reinitialize)
5. REPEAT P = P1 ... PN {
  - VERIFY P = (the new value for the property)
  - CHECK (that the value for P is in use by the network port)

### 7.3.2.X62.1.4 Network Port Configuration Conflict Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that either multiple clients can write to a Network Port object at the same time, or the CONFIGURATION\_IN\_PROGRESS error is reported.

Test Concept: The TD simulates 2 devices (TD and TD2), attempting to write to properties in a Network Port object, O1. The IUT must either accept the second write, or the IUT returns an error with an error class of DEVICE and an error code of CONFIGURATION\_IN\_PROGRESS. Finally, the Network Port's changes are activated and then verified.

1. TRANSMIT WriteProperty-Request,
  - SOURCE = TD,
  - 'Object Identifier' = O1,
  - 'Property Identifier' = (P1: a writable property which utilizes the pending changes functionality),
  - 'Property Value' = (any valid value),
2. RECEIVE BACnet-Simple-ACK
3. TRANSMIT WriteProperty-Request,
  - SOURCE = TD2,
  - 'Object Identifier' = O1,
  - 'Property Identifier' = (P2: a writable property which utilizes the pending changes functionality, and is different than P1, if possible),
  - 'Property Value' = (any valid value),
4. RECEIVE BACnet-Simple-ACK
  - DESTINATION = IUT
  - | BACnet-Error-PDU
    - 'Error Class' = DEVICE,
    - 'Error Code' = CONFIGURATION\_IN\_PROGRESS
5. TRANSMIT ReinitializeDevice-Request,
  - 'Reinitialized State of Device' = ACTIVATE\_CHANGES,
  - 'Password' = (any valid password)
6. RECEIVE BACnet-SimpleACK-PDU
7. MAKE(reconfigure the TD and other devices on the network to the new network settings)
8. WAIT Activate Changes Fail Time
9. IF P1 is a different property than P2 THEN
  - CHECK(P1's value is in use by the network port)
10. CHECK(P2's value is in use by the network port)

### 7.3.2.X62.2 Network-Number-Is Updates Network\_Number\_Quality Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that Network\_Number\_Quality is updated when the IUT learns its Network\_Number from Network-Number-Is.

Test Concept: Write 0 to Network\_Number to set Network\_Number\_Quality to UNKNOWN. Send a Network-Number-Is message to the IUT indicating that the Network\_Number is learned and verify that Network\_Number\_Quality changes to LEARNED. Send a Network-Number-Is message to the IUT indicating that the Network\_Number is configured and verify that Network\_Number\_Quality changes to LEARNED\_CONFIGURED. Write 0 to Network\_Number and verify that Network\_Number\_Quality changes to UNKNOWN.

Test Configuration: Select a Network Port object, O1, which is enabled and has a writable Network\_Number. Connect the TD to the network associated with Network Port O1. This test shall be skipped if the TD cannot be directly connected to the IUT's network.

-- set network number quality to UNKNOWN

1. WRITE Network\_Number = 0
2. TRANSMIT ReinitializeDevice-Request,  
    'Reinitialized State of Device' =        ACTIVATE\_CHANGES,  
    'Password' =                    (any valid password)
3. RECEIVE BACnet-SimpleACK-PDU
4. WAIT Activate Changes Fail Time
5. VERIFY Network\_Number\_Quality = UNKNOWN

-- make IUT learn the network number

6. TRANSMIT Network-Number-Is  
    DESTINATION = LOCAL\_BROADCAST | IUT,  
    'Network Number' = (N1: any valid value)  
    'Flag' = 0                    -- learned
7. VERIFY Network\_Number\_Quality = LEARNED
8. VERIFY Network\_Number = N1

-- make IUT learn the network number from a configure device

9. TRANSMIT Network-Number-Is  
    DESTINATION = LOCAL\_BROADCAST | IUT,  
    'Network Number' = (N2: any valid value)  
    'Flag' = 1                    -- configured
10. VERIFY Network\_Number\_Quality = LEARNED
11. VERIFY Network\_Number = N2

-- configure the IUT's network number

12. WRITE Network\_Number = (N3: any valid value other than 0)
13. TRANSMIT ReinitializeDevice-Request,  
    'Reinitialized State of Device' =        ACTIVATE\_CHANGES,  
    'Password' =                    (any valid password)
14. RECEIVE BACnet-SimpleACK-PDU
15. WAIT Activate Changes Fail Time
16. VERIFY Network\_Number\_Quality = CONFIGURED

17. TRANSMIT Network-Number-Is  
    DESTINATION = LOCAL\_BROADCAST | IUT,  
    'Network Number' = (N4: any valid value)  
    'Flag' = 1                    -- configured
18. VERIFY Network\_Number\_Quality = CONFIGURED
19. VERIFY Network\_Number = N3

-- revert network number quality to UNKNOWN

20. WRITE Network\_Number = 0
21. TRANSMIT ReinitializeDevice-Request,  
    'Reinitialized State of Device' =        ACTIVATE\_CHANGES,  
    'Password' =                    (any valid password)

- 22. RECEIVE BACnet-SimpleACK-PDU
- 23. WAIT Activate Changes Fail Time
- 24. VERIFY Network\_Number\_Quality = UNKNOWN

### 7.3.2.X62.3 Network Port Command Tests

Reason for Change: New test per Addendum 135-2012ai.

#### 7.3.2.X62.3.1 IDLE Command Rejected

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that the Command property does not accept write of IDLE.

Test Concept: Write IDLE to the command property and verify that an error-class of PROPERTY with an error-code of VALUE\_OUT\_OF\_RANGE is returned.

Test Configuration: Execute the test against a Network Port object with a writable Command property. This test shall be skipped if the IUT does not support the Command property. The Network Port object shall have no pending changes.

1. TRANSMIT WriteProperty-Request,
  - 'Object Identifier' = (a Network Port object),
  - 'Property Identifier' = Command,
  - 'Property Value' = IDLE
2. RECEIVE BACnet-Error-PDU
  - 'Error Class' = PROPERTY,
  - 'Error Code' = VALUE\_OUT\_OF\_RANGE

#### 7.3.2.X62.3.2 DISCARD\_CHANGES Command Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that the Network Port discards pending changes when the Command DISCARD\_CHANGES is received.

Test Concept: Write values to one or more properties, P1 .. Px, which utilize the pending changes functionality. Write DISCARD\_CHANGES to the Command property and verify that the properties have reverted to their previous values.

Test Configuration: Execute the test on a Network Port object which supports the DISCARD\_CHANGES command. This test shall be skipped if the IUT does not support the the DISCARD\_CHANGES command.

- save initial values of the properties and change each one to a new value
- 1. REPEAT I = (in the range 1 through the number of properties being written) {
  - V[I] = READ P[I]
  - WRITE P[I] = (a value different than V[I], if possible)
- }
- discard the changes
- 2. WRITE Command = DISCARD\_CHANGES
- 3. WAIT Activate Changes Fail Time
- verify that no changes are pending any more
- 4. VERIFY Changes\_Pending = FALSE
- 5. VERIFY Command = IDLE
- verify that the properties have reverted in value, and that the old value remains in use by the port
- 6. REPEAT I = (in the range 1 through the number of properties being written) {
  - VERIFY P[I] = V[I]
  - CHECK(the value V[I] is in use by the network port)
- }
- command the device to activate any changes which should have no effect

7. TRANSMIT ReinitializeDevice-Request  
     'Reinitialized State of Device' = WARMSTART | ACTIVATE\_CHANGES  
     'Password' = (any valid password)
  8. RECEIVE BACnet-SimpleACK-PDU
  9. MAKE(reconfigure the TD and other devices on the network to the new network settings)
  10. WAIT Activate Changes Fail Time
  11. VERIFY Command = IDLE
- verify that the properties retain their original values, and that that value remains in use by the port
12. REPEAT I = (in the range 1 through the number of properties being written) {  
     VERIFY P[I] = V[I]  
     CHECK(the value V[I] is in use by the network port)  
   }

### 7.3.2.X62.3.3 RENEW\_FD\_REGISTRATION Command Tests

#### 7.3.2.X62.3.3.1 RENEW\_FD\_REGISTRATION Command Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that the Network Port attempts to renew its Foreign Device registration when commanded to do so.

Test Concept: Starting with a Network Port object which is already registered with a BBMD as a Foreign Device, command the Network Port to RENEW\_FD\_REGISTRATION but do not have the TD respond. Verify that the Network Port attempts to renew its Foreign Device registration. Wait until the Network Port has completed its attempt and verify that the Reliability has been set to RENEW\_FD\_REGISTRATION\_FAILURE. Command the Network Port to RENEW\_FD\_REGISTRATION and have the TD respond. Verify that the attempt succeeds and that Reliability is reset to NO\_FAULT\_DETECTED. Command the Network Port to RENEW\_FD\_REGISTRATION and have the TD respond. Verify that the attempt succeeds.

Test Configuration: Configure a Network Port for BACnet/IP or BACnet/IPv6 in FOREIGN mode. Allow the IUT to complete its registration with the TD acting as the BBMD before continuing. If the IUT does not support registering as a Foreign Device, or the IUT does not support the RENEW\_FD\_REGISTRATION command, then this test shall be skipped. The Network Port object shall have no pending changes.

- make sure our initial conditions are good
1. VERIFY Changes\_Pending = FALSE
  2. VERIFY Reliability = NO\_FAULT\_DETECTED
  3. VERIFY BACnet\_IP\_Mode = FOREIGN
- request the renewal, and wait for it to timeout
4. WRITE Command = RENEW\_FD\_REGISTRATION
  5. BEFORE Internal Processing Fail Time  
     RECEIVE Register-Foreign-Device  
     'Time-to-Live' = FD\_Subscription\_Lifetime
  6. WAIT Foreign Device Registration Fail Time
  7. VERIFY Reliability = RENEW\_FD\_REGISTRATION\_FAILURE
  8. VERIFY Command = IDLE
- re-request the renewal, and allow it to succeed
9. WRITE Command = RENEW\_FD\_REGISTRATION
  10. BEFORE Internal Processing Fail Time  
     RECEIVE Register-Foreign-Device  
     'Time-to-Live' = FD\_Subscription\_Lifetime
  11. TRANSMIT BVLC-Result,  
     'Result Code' = Successful completion
  12. VERIFY Reliability = NO\_FAULT\_DETECTED
  13. VERIFY Command = IDLE
  14. WAIT (a random amount of time significantly less than FD\_Subscription\_Lifetime)



- re-request the renewal, and allow it to succeed
- 15. WRITE Command = RENEW\_FD\_REGISTRATION
- 16. BEFORE Internal Processing Fail Time
  - RECEIVE Register-Foreign-Device
  - 'Time-to-Live' = FD\_Subscription\_Lifetime
- 17. TRANSMIT BVLC-Result,
  - 'Result Code' = Successful completion
- 18. VERIFY Reliability = NO\_FAULT\_DETECTED
- 19. VERIFY Command = IDLE

### 7.3.2.X62.3.3.2 RENEW\_FD\_REGISTRATION Command Failure Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that Network Port object respond to RENEW\_FD\_REGISTRATION commands when the command is not supported / enabled.

Test Concept: Attempt to command a Network Port which is either not an BACnet/IP nor BACnet/IPv6 port or which is not in FOREIGN mode, to renew its FD subscription. Verify that the attempt fails with an error class of PROPERTY and an error code of VALUE\_OUT\_OF\_RANGE.

Test Configuration: Select a Network Port which is not in FOREIGN mode. If the IUT does not support the Command property, then this test shall be skipped.

- make sure our initial conditions are good
- 1. IF Network\_Type is IPV4 or IPV6 THEN
- 2. VERIFY BACnet\_IP\_Mode <> FOREIGN
  
- 3. TRANSMIT WriteProperty-Request,
  - 'Object Identifier' = (the Network Port object),
  - 'Property Identifier' = Command,
  - 'Property Value' = RENEW\_FD\_SUBSCRIPTION,
- 4. RECEIVE BACnet-Error-PDU
  - 'Error Class' = PROPERTY,
  - 'Error Code' = VALUE\_OUT\_OF\_RANGE
- 5. VERIFY Command = IDLE

### 7.3.2.X62.3.4 RESTART\_SLAVE\_DISCOVERY Command Tests

#### 7.3.2.X62.3.4.1 RESTART\_SLAVE\_DISCOVERY Command Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that the Network Port restarts the slave discovery process when commanded to.

Test Concept: Starting with a Network Port object which is configured as an MS/TP Slave Proxy, command the Network Port to RESTART\_SLAVE\_DISCOVERY. Verify that the IUT restarts slave discovery.

Test Configuration Configure a Network Port object which is for MS/TP and as an MS/TP Slave Proxy. Configure the TD to act as an MS/TP slave. Delay the stat of the test until after the IUT has completed its initial slave confirmation. If the IUT does not support registering as a Foreign Device, or the IUT does not support the RENEW\_FD\_REGISTRATION command, then this test shall be skipped.

- make sure our initial conditions are good
- 1. VERIFY Network\_Type = MSTP
- 2. VERIFY Slave\_Proxy\_Enable = TRUE
  
- request the renewal, and wait for it to timeout
- 3. WRITE Command = RESTART\_SLAVE\_DISCOVERY
- 4. BEFORE Slave Proxy Confirm Interval

```

REPEAT addr=(all MS/TP addresses excluding the IUT's MAC address) DO {
  RECEIVE DESTINATION=addr, SRC=IUT
  ReadProperty-Request,
  'Object Identifier' = (DEVICE,4194303),
  'Property Identifier' = Protocol_Services_Supported
}

```

5. VERIFY Command = IDLE

Notes to Tester: The IUT may interrogate the slave addresses in any order. The IUT is allowed to generate any other traffic during the test including, and is not limited to reading property values from the devices it finds.

### 7.3.2.X62.3.4.2 RESTART\_SLAVE\_DISCOVERY Command Failure Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that Network Port object respond to RESTART\_SLAVE\_DISCOVERY commands when the command is not supported / enabled.

Test Concept: Attempt to command a Network Port which is not acting as a slav proxy to RESTART\_SLAVE\_DISCOVERY. Verify that the attempt fails with an error class of PROPERTY and an error code of VALUE\_OUT\_OF\_RANGE.

Test Configuration: Select a Network Port which is not configured to be a slave proxy. If the IUT supports slave proxy functionality, this test shall be skipped as the standard does not specify how the IUT should respond when slave proxy is supported but not enabled. If the IUT does not support the Command property, then this test shall be skipped.

```

1. TRANSMIT WriteProperty-Request,
   'Object Identifier' = (the Network Port object),
   'Property Identifier' = Command,
   'Property Value' = RESTART_SLAVE_DISCOVERY
2. IF Network_Type is MSTP THEN
   RECEIVE BACnet-Error-PDU
   'Error Class' = PROPERTY,
   'Error Code' = OPTIONAL_FUNCTIONALITY_SUPPORTED
ELSE
   RECEIVE BACnet-Error-PDU
   'Error Class' = PROPERTY,
   'Error Code' = VALUE_OUT_OF_RANGE
3. VERIFY Command = IDLE

```

### 7.3.2.X62.3.5 RENEW\_DHCP Command Tests

#### 7.3.2.X62.3.5.1 RENEW\_DHCP Command Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that the Network Port attempts to renew its addressing infomation when commanded to.

Test Concept: Starting with a Network Port object which is configured to use DHCP and which supports the RENEW\_DHCP command, command the port to RENEW\_DHCP. Verify that the IUT requests a renewal of addressing information.

Test Configuration Configure a Network Port object which is for MS/TP and as an MS/TP Slave Proxy. Configure the TD to act as an MS/TP slave. Delay the stat of the test until after the IUT has completed its initial slave confirmation. If the IUT does not support registrering as a Foreign Device, or the IUT does not support the RENEW\_FD\_REGISTRATION command, then this test shall be skipped.

-- make sure our initial conditions are good

```

1. IF Network_Type = IPV4 THEN
   VERIFY IP_DHCP_Enable = TRUE
2. IF Network_Type = IPV6 THEN
   VERIFY IPv6_Auto_Addresssing_Enable = TRUE

```

- request the renewal, and wait for it to timeout
- 3. WRITE Command = RENEW\_DHCP
- 4. CHECK(that the IUT requested a renewal of its addressing information)
- 5. VERIFY Command = IDLE

### 7.3.2.X62.3.5.2 RENEW\_DHCP Command Failure Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that Network Port object respond to RENEW\_DHCP commands when the command is not supported / enabled.

Test Concept: Attempt to command a Network Port which is either not an BACnet/IP nor BACnet/IPv6 port or which is not in configured for auto-addressing. Verify that the attempt fails with an error class of PROPERTY and an error code of VALUE\_OUT\_OF\_RANGE.

Test Configuration: Select a Network Port which is not an IP or IPV6 port setup for autoaddressing. If the IUT does not support the Command property, then this test shall be skipped.

- make sure our initial conditions are good
- 1. IF Network\_Type is IPV4 and IP\_DHCP\_Enable is present THEN  
    VERIFY IP\_DHCP\_ENABLE = FALSE
- 2. IF Network\_Type is IPV6 and IPV6\_Auto\_Addressing\_Enabled is present THEN  
    VERIFY IPV6\_Auto\_Addressing\_Enabled = FALSE
- 3. TRANSMIT WriteProperty-Request,  
    'Object Identifier' = (the Network Port object),  
    'Property Identifier' = Command,  
    'Property Value' = RENEW\_DHCP
- 4. IF Network\_Type is IPV4 or IPV6 THEN  
    RECEIVE BACnet-Error-PDU  
    'Error Class' = PROPERTY,  
    'Error Code' = OPTIONAL\_FUNCTIONALITY\_NOT\_SUPPORTED  
    ELSE  
    RECEIVE BACnet-Error-PDU  
    'Error Class' = PROPERTY,  
    'Error Code' = VALUE\_OUT\_OF\_RANGE
- 5. VERIFY Command = IDLE

### 7.3.2.X62.3.6 RESTART\_AUTONEGOTIATION Command Tests

#### 7.3.2.X62.3.6.1 RESTART\_AUTONEGOTIATION Command Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that the Network Port attempts to re-autonegotiate its link speed when commanded to.

Test Concept: Starting with a Network Port object which is configured to auto-negotiate its link speed and which supports the RESTART\_AUTONEGOTIATION command, is commanded to restart autonegotiation. The link speed is changed, and it is verified that the IUT performs link speed negotiation and is able to communicate with the new speed.

Test Configuration: The TD and IUT are connected on a network for which the IUT performs link speed auto negotiation. The Network Port object for the port is configured to perform auto-negotiation, If the IUT does not support the RESTART\_AUTONEGOTIATION command, then this test shall be skipped.

- make sure our initial conditions are good
- 1. VERIFY Link\_Speed\_Autonegotiate = TRUE

- request the renewal, and wait for it to timeout
- 3. WRITE Command = RESTART\_AUTONEGOTIATION
- 4. MAKE(change the link speed for the network or link)
- 5. WAIT Auto Negotiation Fail Time
- 6. CHECK(check any external indications that the new link speed was detected)
- 7. VERIFY Command = IDLE
  - the act of validating the command
  - property is sufficient to validate that the command worked

### 7.3.2.X62.3.6.2 RESTART\_AUTONEGOTIATION Command Failure Test

Reason for Change: New test per Addendum 135-2012ai.

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

Test Concept: Starting with a Network Port object which is not configured to auto-negotiate its link speed or which does not support the RESTART\_AUTONEGOTIATION, command it to restart autonegotiation. Verify that the correct error code is returned.

Test Configuration: If the network port support auto-negotiation, disable it. If the IUT does not support the Command property, or all Network Port object support auto-negotiation and it can't be disabled, then this test shall be skipped.

- make sure our initial conditions are good
- 1. VERIFY Link\_Speed\_Autonegotiate = TRUE
- request the renewal, and wait for it to timeout
- 2. TRANSMIT WriteProperty-Request,
  - 'Object Identifier' = (the Network Port object),
  - 'Property Identifier' = Command,
  - 'Property Value' = RESTART\_AUTONEGOTIATION
- 3. IF the port does not support autonegotiation THEN
  - RECEIVE BACnet-Error-PDU
    - 'Error Class' = PROPERTY,
    - 'Error Code' = OPTIONAL\_FUNCTIONALITY\_NOT\_SUPPORTED
  - ELSE
    - RECEIVE BACnet-Error-PDU
      - 'Error Class' = PROPERTY,
      - 'Error Code' = VALUE\_OUT\_OF\_RANGE

### 7.3.2.X62.3.7 DISCONNECT Command Tests

#### 7.3.2.X62.3.7.1 DISCONNECT Command Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that the Network Port attempts to disconnect its link speed when commanded to.

Test Concept: Starting with a Network Port object which supports the DISCONNECT command. The port is commanded to disconnect. The disconnection of the link is verified.

Test Configuration: The TD and IUT are connected on a network which supports disconnection. If the IUT does not support the DISCONNECT command, then this test shall be skipped.

- make sure our initial conditions are good
- 1. VERIFY Network\_Type = (a network type that supports disconnection, such as PTP)
- 2. WRITE Command = DISCONNECT
- 3. CHECK(that the link was disconnected)

### 7.3.2.X62.3.7.2 DISCONNECT Command Failure Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that Network Port objects respond to the DISCONNECT 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 disconnection, command it to disconnect. Verify that the correct error code is returned.

Test Configuration: If the IUT does not support the Command property, or all Network Port object support disconnection then this test shall be skipped.

1. TRANSMIT WriteProperty-Request,  
'Object Identifier' = (the Network Port object),  
'Property Identifier' = Command,  
'Property Value' = DISCONNECT
2. RECEIVE BACnet-Error-PDU  
'Error Class' = PROPERTY,  
'Error Code' = VALUE\_OUT\_OF\_RANGE

### 7.3.2.X62.3.8 RESTART\_PORT Command Tests

#### 7.3.2.X62.3.8.1 RESTART\_PORT Command Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that the Network Port attempts to restart its port when commanded to.

Test Concept: With a Network Port object which supports the RESTART\_PORT command, command the port to restart. The restart of the port is verified.

Test Configuration: If the IUT does not support the RESTART\_PORT command, then this test shall be skipped.

-- make sure our initial conditions are good

1. WRITE Command = RESTART\_PORT
2. CHECK(that the port was restarted)

#### 7.3.2.X62.3.8.2 RESTART\_PORT Command Failure Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that Network Port objects respond to the RESTART\_PORT command with the correct error codes when the command is not supported.

Test Concept: With a Network Port object which does not support the RESTART\_PORT command, command the port to restart. Verify that the correct error code is returned.

Test Configuration: If the IUT does not support the Command property, or all Network Port object support disconnection then this test shall be skipped.

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

### 7.3.2.X62.3.9 No Commands if Changes\_Pending Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that the Network Port disallows commands, except DISCARD\_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.

Test Configuration: 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
3. REPEAT CMD = (all valid values that NP supports except DISCARD_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)
}
```

### 7.3.2.X62.4 Hierarchical Network Port Tests

#### 7.3.2.X62.4.1 Valid Hierarchy Test

Reason for Change: New test per Addendum 135-2012ai.

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 Network Port object which has a Protocol_Level of
    BACNET_APPLICATION or NON_BACNET_APPLICATION) {
2. REPEAT NPx = (object id of each Network Port object 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
    }
}
}
```

#### 7.3.2.X62.4.2 Properties in Referenced Network Port Reflected in Top Network Port Object

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that properties in referenced Network Port objects are reflected in the top Network Port object.

Test Concept: Visit each Network Port object which represents a configured BACnet application layer port. Visit each Network Port object in the hierarchy ensuring that the properties in the referenced Network Port object exist and have the same value in the top Network Port object.

1. REPEAT NP = (object id of each Network Port object which has a Protocol\_Level of BACNET\_APPLICATION) {
  - verify that the required properties exist for this Network Port object based
  - on its Network\_Type
2. REPEAT P = (each required property for NP's Network\_Type, see Table 12-72) {
  - VERIFY (Network Port, NP), P = (any valid value)
3. REPEAT NPx = (object id of each Network Port object in NP's hierarchy) {
  - verify that the expected properties exist in the Network Port object based
  - on its Network\_Type and Protocol\_Level. In addition, verify that the property
  - value is inherited into NP (unless already inherited from a different Network Port)
  - REPEAT P = (each expected property in NPx based on its Network\_Type and Protocol\_Level as defined in Table 12-73) {
    - V1 = READ (Network Port, NPx), P
    - IF P is not in a higher Network Port object in this hierarchy THEN
    - VERIFY (Network Port, NP), P = V1

#### 7.3.2.X62.4.3 Changes Reflected in Top Network Port Object

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that changing properties in child Network Port objects result in the new property values reflected in the top Network Port object.

Test Concept: Write a writable, inheritable property within a Network Port's hierarchy and verify that the new value is reflected in the top Network Port object after activating the change, if required.

Test Configuration: Select a Network Port object, O1, which represents a configured network port, has a Protocol\_Level of BACNET\_APPLICATION, which references a Network Port object and for which there is a writable inherited property, P, within hierarchy. Let O2 be the Network Port object which contains P.

1. V1 = READ O2, P
2. VERIFY O1, P = V1
3. WRITE O2, P = (V2: any valid value different that V1)
4. IF O2, Changes\_Pending THEN
  - TRANSMIT ReinitializeDevice
  - 'Reinitialized State of Device' = ACTIVATE\_CHANGES
  - RECEIVE BACnet-SimpleACK-PDU
  - MAKE(reconfigure the TD and other devices on the network to the new network settings)
  - WAIT Activate Changes Fail Time
5. VERIFY O1, P = V2

#### 7.3.2.X62.4.4 Changes Reflected in Lower Network Port Objects

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that changing properties in the top Network Port object results in the new property values reflected in the child Network Port objects.

Test Concept: Write a writable, inherited property within a top Network Port object and verify that the new value is reflected in the property from which value is inherited after activating the change, if required.

Test Configuration: Select a Network Port object, O1, which represents a configured network port, has a Protocol\_Level of BACNET\_APPLICATION, which references a Network Port object and for which there is a writable inherited property, P, in O1 which inherits its value from property P in O2.

1. V1 = READ O2, P
2. VERIFY O1, P = V1
3. WRITE O1, P = (V2: any valid value different than V1)
4. IF O1, Changes\_Pending THEN
  - TRANSMIT ReinitializeDevice
  - 'Reinitialized State of Device' = ACTIVATE\_CHANGES
  - RECEIVE BACnet-SimpleACK-PDU
  - MAKE(reconfigure the TD and other devices on the network to the new network settings)
  - WAIT Activate Changes Fail Time
5. VERIFY O2, P = V2

### 7.3.2.X62.5 APDU\_Length Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that the Device objects report a Max\_APDU\_Length\_Accepted that is not larger than the largest value reported by the configured and enabled Network Port objects.

Test Concept: Determine the largest APDU\_Length property for all configured and enabled Network Port objects with a Protocol\_Level of BACNET\_APPLICATION. Verify that each is larger than 50 and less than or equal the maximum allowed for the attached datalink. Verify that the Max\_APDU\_Length\_Supported property of the Device object is not larger than that maximum.

1. MAX\_APDU = 65536
2. REPEAT NP = (all configured and enabled Network Port objects with a Protocol\_Level of BACNET\_APPLICATION) {
  - IF NP.APDU\_Length < 50 THEN
  - ERROR "APDU\_Length must not be less than 50."
  - IF NP.APDU\_Length > (the maximum allowable for the Network\_Type) THEN
  - ERROR "APDU\_Length is too large for the connected Network\_Type"
  - IF MAX\_APDU > NP.APDU\_Length THEN
  - MAX\_APDU = NP.APDU\_Length
3. VERIFY (Device, 4194303), Max\_APDU\_Length\_Supported <= MAX\_APDU

### 7.3.2.X62.6 Routing\_Table Test

Reason for Change: New test per Addendum 135-2012ai.

Purpose: To verify that routes are added to the Routing\_Table property of the Network Port object when they are found.

Test Concept: Starting with a clear routing table, send an I-Am-Router-To-Network message with multiple networks listed and verify that all are added to the Routing\_Table. Send the remaining Nmax-2 I-Am-Router-To-Network messages to the IUT and verify that the entries are placed into the Routing\_Table. Verify that no other Network Port objects are affected by the messages.

Configuration: All enabled Network Port objects, NP1 .. NPx, have empty Routing\_Table properties. NP1 is the Network Port for port A. Nmax is the smaller of the maximum number of entries the IUT can hold in its routing table, and the maximum that can be encoded in a single segment ReadProperty response. N1 .. Nmax is a set of random network numbers, none of which are in use by the IUT. R1 .. Rmax are the router MAC addresses for each of the network numbers in N1 .. Nmax. R1 and R2 shall be the same. The TD and IUT shall be on the same BACnet network and there shall be no other routers connected.

-- verify that no routes are known

1. REPEAT NP = (all enabled Network Port objects with a Protocol\_Level of BACNET\_APPLICATION) {
  - VERIFY NP, Routing\_Table = ()

-- verify that the IUT notices all routes in the I-Am-Router-To-Network



2. IF the IUT supports storing the address of more than 1 router THEN

```

TRANSMIT PORT A
  DESTINATION = LOCAL BROADCAST,
  SOURCE R1,
  I-Am-Router-To-Network,
  Network Numbers = N1, N2
  VERIFY NP1, Routing_Table = (
    (N1, R1, AVAILABLE, (optionally, any valid index)),
    (N2, R2, AVAILABLE, (optionally, any valid index))
  ) -- the order of the entries does not matter

```

ELSE

```

TRANSMIT PORT A
  DESTINATION = LOCAL BROADCAST,
  SOURCE R1,
  I-Am-Router-To-Network,
  Network Numbers = N1
  VERIFY NP1, Routing_Table = (
    (N1, R1, AVAILABLE, (optionally, any valid index)),
  )

```

-- verify that the IUT supports up to Nmax entries

```

4. REPEAT NP = (all enabled Network Port objects, except NP1, with a Protocol_Level of
  BACNET_APPLICATION) {
  VERIFY NP, Routing_Table = ()
}

```

```

5. REPEAT N,R = (N2 up to Nmax, R2 up to Rmax) {
  TRANSMIT PORT A
  DESTINATION = LOCAL BROADCAST,
  SOURCE = R
  I-Am-Router-To-Network,
  Network Numbers = N
}

```

```

6. VERIFY NP1, Routing_Table = (
  N1, R1, AVAILABLE, (optionally, any valid index),
  N2, R2, AVAILABLE, (optionally, any valid index),
  ...
  Nmax, Rmax, AVAILABLE, (optionally, any valid index)
) -- the order of the entries does not matter

```

-- verify that the other Network Port objects are unaffected

```

7. REPEAT NP = (all enabled Network Port objects, except NP1, with a Protocol_Level of
  BACNET_APPLICATION) {
  VERIFY NP, Routing_Table = ()
}

```

Notes to Tester: If the network cannot be configured with the TD and the IUT on the same network, the test shall be adjusted to include the router to the TDs network instead of having a cleared routing table from the start of the test.

[Insert new test 7.3.2.X62.7]

### 7.3.2.X62.7 DHCP Tests

#### 7.3.2.X62.7.1 Basic IP DHCP Test

Purpose: Verify that the IUT is able to participate in IPv4 DHCP and correctly report its DHCP status.

Test Concept: The DHCP server is removed from network. The IUT is is the configured with an IPv4 network requiring DHCP, and if required, its DHCP settings are cleared. The related Network Port object is queried to verify that the DHCP related properties have the appropriate values indicating DHCP has not completed. The DHCP is connected to the network and the

network. It is verified that the IUT obtains network settings from the DHCP server, and that the DHCP properties reflect the current status.

Configuration Requirements: The DHCP is disconnected from the network or turned off. The IUT is configured for DHCP and any settings it previously received via DHCP are cleared.

1. IF the IUT has a second enabled network port THEN
  - VERIFY IP\_DHCP\_Enable = True
  - IF IP\_DHCP\_Lease\_Time property is present THEN
    - VERIFY IP\_DHCP\_Lease\_Time = 0
  - IF IP\_DHCP\_Lease\_Time\_Remaining property is present THEN
    - VERIFY IP\_DHCP\_Lease\_Time\_Remaining = 0
  - IF IP\_DHCP\_Server property is present THEN
    - VERIFY IP\_DHCP\_Server = X'00000000'
2. MAKE(connect the DHCP server to the network)
3. WAIT until the IUT obtains DHCP information
4. IF IP\_DHCP\_Lease\_Time property is present THEN
  - VERIFY IP\_DHCP\_Lease\_Time = (0 or the value provided by the DHCP server)
5. IF IP\_DHCP\_Lease\_Time\_Remaining property is present THEN
  - VERIFY IP\_DHCP\_Lease\_Time\_Remaining = (0 or a value less than that provided by the DHCP server)
6. IF IP\_DHCP\_Server property is present THEN
  - VERIFY IP\_DHCP\_Server = (the DHCP server's address of X'00000000')

### 7.3.2.X62.7.2 Basic IPv6 DHCP Test

Purpose: Verify that the IUT is able to participate in IPv6 DHCP and correctly report its DHCP status.

Test Concept: The DHCP server is removed from network. The IUT is is the configured with an IPv6 network requiring DHCP, and if required, its DHCP settings are cleared. The related Network Port object is queried to verify that the DHCP related properties have the appropriate values indicating DHCP has not completed. The DHCP is connected to the network and the network. It is verified that the IUT obtains network settings from the DHCP server, and that the DHCP properties reflect the current status.

Configuration Requirements: The DHCP is disconnected from the network or turned off. The IUT is configured for DHCP and any settings it previously received via DHCP are cleared.

1. IF the IUT has a second enabled network port THEN
  - VERIFY IPv6\_DHCP\_Enable = True
  - IF IPv6\_DHCP\_Lease\_Time property is present THEN
    - VERIFY IPv6\_DHCP\_Lease\_Time = 0
  - IF IPv6\_DHCP\_Lease\_Time\_Remaining property is present THEN
    - VERIFY IPv6\_DHCP\_Lease\_Time\_Remaining = 0
  - IF IPv6\_DHCP\_Server property is present THEN
    - VERIFY IPv6\_DHCP\_Server = X'00000000'
2. MAKE(connect the DHCP server to the network)
3. WAIT until the IUT obtains DHCP information
4. IF IPv6\_DHCP\_Lease\_Time property is present THEN
  - VERIFY IPv6\_DHCP\_Lease\_Time = (0 or the value provided by the DHCP server)
5. IF IPv6\_DHCP\_Lease\_Time\_Remaining property is present THEN
  - VERIFY IPv6\_DHCP\_Lease\_Time\_Remaining = (0 or a value less than that provided by the DHCP server)
6. IF IPv6\_DHCP\_Server property is present THEN
  - VERIFY IPv6\_DHCP\_Server = (the DHCP server's address of X'00000000')

[Insert clause 7.3.1.1, rename and move existing 7.3.1.1 to 7.3.1.1.1]

[Modify all references to 7.3.1.1 to 7.3.1.1.1 in BTL Test Plan]

#### 7.3.1.1 *Out\_Of\_Service, Status\_Flags, and Reliability Tests*

##### 7.3.1.1.1 ~~7.3.1.1~~ *Out\_Of\_Service, Status\_Flags, and Reliability Test*

[Insert new test 7.3.1.1.2]

### 7.3.1.1.2 Out\_Of\_Service, Status\_Flags, and Reliability Test for Objects without Present\_Value

Purpose: This test verifies the interrelationship between the Out\_Of\_Service, Status\_Flags, and Reliability properties. If the PICS indicates that the Out\_Of\_Service property of the object under test is not writable, and if the value of the property cannot be changed by other means, then this test shall be omitted. This test applies to objects that do not contain Present\_Value.

Test Concept: Write to and verify the interrelationship between the Out\_Of\_Service, Status\_Flags, and Reliability properties of an object which does not contain Present\_Value.

Configuration Requirements: The selected object is configured such that its Reliability is NO\_FAULT\_DETECTED before execution of this test.

Test Steps:

1. IF (Out\_Of\_Service is writable) THEN  
     WRITE Out\_Of\_Service = TRUE  
   ELSE  
     MAKE (Out\_Of\_Service = TRUE)
2. VERIFY Out\_Of\_Service = TRUE
3. VERIFY Status\_Flags = (?, FALSE, ?, TRUE)
4. IF (Reliability is present and writable) THEN  
     REPEAT X = (all values of the Reliability enumeration appropriate to the object type except  
                 NO\_FAULT\_DETECTED) DO {  
         WRITE Reliability = X  
         VERIFY Reliability = X  
         VERIFY Status\_Flags = (TRUE, TRUE, ?, TRUE)  
         WRITE Reliability = NO\_FAULT\_DETECTED  
         VERIFY Reliability = NO\_FAULT\_DETECTED  
         VERIFY Status\_Flags = (?, FALSE, ?, TRUE)  
     }
5. CHECK (functionality that should stop or be disabled is. For example, with a Network Port object, all communication of the protocol modeled by the object, through that port is disabled)
6. IF (Out\_Of\_Service is writable) THEN  
     WRITE Out\_Of\_Service = FALSE  
   ELSE  
     MAKE (Out\_Of\_Service = FALSE)
7. VERIFY Out\_Of\_Service = FALSE
8. VERIFY Status\_Flags = (?, ?, ?, FALSE)

[ Add Clause 9.18.1.X5 ]

### 9.18.1.X5 ReadProperty of the Network Port Object using the Unknown Instance

Purpose: Verify that the IUT selects the correct object when a Network Port is read using the special object instance 4194303.

Test Concept: Execute a ReadProperty service request specifying 'Object Identifier' = (Network Port, 4194303). Verify that the responding BACnet-user selects the local Network Port object representing the network port through which the request was received.

Configuration Requirements: Let X be the instance number of the Network Port object associated with the network port through which the TD will communicate with the IUT.

Test Steps:

1. TRANSMIT ReadProperty-Request,  
     'Object Identifier' = (Network Port, 4194303),  
     'Property Identifier' = Object-Identifier

2. RECEIVE ReadProperty-ACK,  
     'Object Identifier' = (Network Port, X),  
     'Property Identifier' = Object-Identifier,  
     'Property Value' = (Network Port, X)
3. REPEAT P = (each property in the specified Network Port object) {  
     TRANSMIT ReadProperty-Request through the same port as above,  
     'Object Identifier' = (Network Port, 4194303),  
     'Property Identifier' = P  
     RECEIVE ReadProperty-ACK,  
     'Object Identifier' = (Network Port, X),  
     'Property Identifier' = P,  
     'Property Value' = V  
     VERIFY (Network Port, X), P = V  
   }

[ Add Clause 9.20.1.X3 ]

### 9.20.1.X3 ReadPropertyMultiple of the Network Port Object using the Unknown Instance

Purpose: Verify that the IUT selects the correct object when a Network Port is read using the special object instance 4194303.

Test Concept: Execute a ReadPropertyMultiple service request specifying 'Object Identifier' = (Network Port, 4194303). The responding BACnet-user selects the local Network Port object representing the network port through which the request was received.

Configuration Requirements: Let X be the instance number of the Network Port object associated with the network port through which the TD will communicate with the IUT.

Test Steps:

1. TRANSMIT ReadPropertyMultiple-Request,  
     'Object Identifier' = (Network Port, 4194303),  
     'Property Identifier' = Object-Identifier
2. RECEIVE ReadPropertyMultiple-ACK,  
     'Object Identifier' = (Network Port, X),  
     'Property Identifier' = Object-Identifier,  
     'Property Value' = (Network Port, X)
3. REPEAT P = (each property in the specified Network Port object) {  
     TRANSMIT ReadPropertyMultiple-Request through the same port as above,  
     'Object Identifier' = (Network Port, 4194303),  
     'Property Identifier' = P  
     RECEIVE ReadPropertyMultiple-ACK,  
     'Object Identifier' = (Network Port, X),  
     'Property Identifier' = P,  
     'Property Value' = V  
     VERIFY (Network Port, X), P = V  
   }

[In BTL Specified Tests, add new test]

### 14.3.X1 Write-BDT service is required to return Write-BDT-NAK

Reason for Change: Clause J.4.4.2 mandates a change and that all devices claiming Protocol\_Revision >= 17, shall behave in this changed way.

Purpose: To verify that any IUT with Protocol\_Revision claimed as 17 or higher, will return Write-Broadcast-Distribution-Table NAK to every Write-Broadcast-Distribution-Table request.

Configuration Requirements: If the Protocol\_Revision claimed is less than 17, this test shall be skipped.





*VERIFY NP, BBMD\_Foreign\_Device\_Table = ( (B/IP address of FD2, 0, 20 - execution time) )*

7. WAIT (30 seconds)

8. TRANSMIT

DA = IUT,

SA = FD2,

Read-Foreign-Device-Table

9. RECEIVE

DA = FD2,

SA = IUT,

Read-Foreign-Device-Table-Ack,

(No FDT entries)

10. *IF Protocol\_Revision >= 17 THEN*

*VERIFY NP, BBMD\_Foreign\_Device\_Table = ( )*

Note to tester: The accuracy of the FDT timer shall be specified by the vendor.

**BTL-16.1ai-2: DM-RD-B Negative Testing - BTLWG-92**

**Overview:**

DM-RD-B Negative Test.

**Changes:**

**BTL Checklist Changes**

None

**BTL Test Plan Changes**

[In BTL Test Plan, add an entry into 8 Device Management - Reinitialize Device -B section]

**8.16 Device Management - Reinitialize Device - B**

**8.16.1 Base Requirements**

Base requirements must be met by any IUT claiming conformance to this BIBB. There are no base requirements tests for this section.

<b>BTL - 9.27.2.X Rejects Unsupported Reinitialize Types</b>	
<b>Test Conditionality</b>	If the IUT supports all values for the 'Reinitialized State of Device' parameters, this test shall be skipped.
<b>Test Directives</b>	If the device does not support DM-BR-B, then all values related to Backup and Restore shall be tested. If the device does not support activation of Network Port changes, ACTIVATE_CHANGES shall be tested.
<b>Testing Hints</b>	

**BTL Specified Tests Changes**

[Add in BTL Specified Tests, a new ReinitializeDevice negative test]

**9.27.2.X Rejects Unsupported Reinitialize Types**

Reason For Change: Added to ensure that non-supported 'Reinitialized State of Device' are properly rejected.

Purpose: Verify that IUT correctly rejects unsupported 'Reinitialized State of Device' values.

Test Concept: Send each unsupported 'Reinitialized State of Device' value to the device and ensure that it correctly rejects the value.

Test Steps:

1. REPEAT S = (each unsupported 'Reinitialized State of Device' value) {
  - TRANSMIT ReinitializeDevice
  - 'Reinitialized State of Device' = S,
  - 'Password' = (any valid value)
  - RECEIVE BACnet-Error-PDU
  - 'Error Class' = SERVICES,
  - 'Error Code' = VALUE\_OUT\_OF\_RANGE |
  - OPTIONAL\_FUNCTIONALITY\_NOT\_SUPPORTED