BACnet® TESTING LABORATORIES ADDENDA

Addendum cr-imp1 to BTL Test Package 26.1

Revision final
Revised 6/29/2026

Approved by the BTL Working Group on April 27, 2026;
Approved by the BTL Working Group Voting Members on June 22, 2026
Published on July 3, 2026

BACnet is a registered trademark of ASHRAE. ASHRAE does not endorse, approve or test products for compliance with ASHRAE standards. Compliance of listed products to the requirements of ASHRAE Standard 135 is the responsibility of BACnet International. BTL is a registered trademark of BACnet International.

[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-26.1 cr-imp1-1: Exception Schedule Add Testing When 255 Entries [BTLWG-966, CR-0266].....2
BTL-26.1 cr-imp1-2: Update DM-DCC and DM-RD B-side Testing [BTLWG-1726, CR-0582] 5
BTL-26.1 cr-imp1-3: Initiate ReadRange Test Conditionality [BTLWG-1748, CR-0592].....18
BTL-26.1 cr-imp1-4: Network Port - Unknown and Unsupported Commands [BTLWG-1767, CR-0596].....20
BTL-26.1 cr-imp1-5: Malformed BVLC Test for Nodes operating as a Hub Issues [BTLWG-1782, CR-0598] 22
BTL-26.1 cr-imp1-6: Update COV Subscription Support Testing [BTLWG-1801, CR-0604] 26
BTL-26.1 cr-imp1-7: Initiating You-Are Messages and User Actions [BTLWG-1804, CR-0603] 28
BTL-26.1 cr-imp1-8: Improve Testing of Writing Stream Based File [BTLWG-1794, CR-0601].....32
BTL-26.1 cr-imp1-9: Wrong Message Id Requirement in Heartbeat Tests [BTLWG-1822, CR-0607] 34
BTL-26.1 cr-imp1-10: Update Test Directives for Router Tests [BTLWG-932, CR-0215] 39

In the following document, language to be added to existing clauses within the BTL Test Package 26.1 is indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are proposed to be added, plain type is used throughout.

In contrast, changes to BTL Specified Tests also contain a yellow highlight to indicate the changes made by this addendum. When this addendum is applied, all highlighting will be removed. Change markings on tests will remain to indicate the difference between the new test and an existing 135.1 test. If a test being modified has never existed in 135.1, the applied result should not contain any change markings. When this is the case, square brackets will be used to describe the changes required for this test.

Each addendum can stand independently unless specifically noted via dependency within the addendum. If multiple addenda change the same test or section, each future released addendum that changes the same test or section will note in square brackets whether or not those changes are reflected.

© 2026 by BACnet International. All rights reserved.

BTL-26.1 cr-imp1-1: Exception Schedule Add Testing When 255 Entries [BTLWG-966, CR-0266]

Overview:

The 135.1 test standard largely tries to avoid any requirement of running tests in a sequence. As per test step 1 of tests 13.10.4.7, 13.10.4.8, 13.10.4.9, 13.10.4.10, condition shall be made so that IUT adds an entire BACnetSpecialEvent to the Exception_Schedule of S1 in TD. As per Configuration of S1, it already contains 255 BACnetSpecialEvent and trying to add one new BACnetSpecialEvent, TD will give value-out-of-range error.

Therefore, before execution of tests 13.10.4.7, 13.10.4.8, 13.10.4.9, 13.10.4.10, there is a need of execution of test 13.10.4.6, wherein IUT is deleting an entire BACnetSpecialEvent from the Exception_Schedule of S1 in TD.

Changes:

Checklist Changes

None

Test Plan Changes

~~135.1-2025~~ BTL- 13.10.4.7 - Modify an Exception_Schedule by adding a BACnetSpecialEvent with period of choice calendarEntry of choice Date		
	Test Conditionality	The reference server schedule (~~S1~~*S4*) shall contain BACnetTimeValue of type Enumerated.
	Test Directives	
	Testing Hints	The tester should attempt to add an entire BACnetSpecialEvent of choice Date to the reference schedule.
~~135.1-2025~~ BTL- 13.10.4.8 - Modify an Exception_Schedule by adding a BACnetSpecialEvent with period of choice calendarEntry of choice DateRange		
	Test Conditionality	The reference server schedule (~~S1~~*S4*) shall contain BACnetTimeValue of type Enumerated.
	Test Directives	
	Testing Hints	The tester should attempt to add an entire BACnetSpecialEvent of choice DateRange to the reference schedule.
~~135.1-2025~~ BTL- 13.10.4.9 - Modify an Exception_Schedule by adding a BACnetSpecialEvent with Period of Choice calendarEntry of Choice WeekNDay		
	Test Conditionality	The reference server schedule (~~S1~~*S4*) shall contain BACnetTimeValue of type Enumerated.
	Test Directives	
	Testing Hints	The tester should attempt to add an entire BACnetSpecialEvent of choice WeekNDay to the reference schedule.
~~135.1-2025~~ BTL- 13.10.4.10 - Modify an Exception_Schedule by adding a BACnetSpecialEvent with period of choice calendarReference		
	Test Conditionality	The reference server schedule (~~S1~~*S4*) shall contain BACnetTimeValue of type Enumerated.
	Test Directives	
	Testing Hints	The tester should attempt to add an entire BACnetSpecialEvent of choice calendarReference to the reference schedule.

Specified Test Changes

© 2026 by BACnet International. All rights reserved.

[Add new sentences to section 13.10, introducing new S4 reference schedule]

13.10 Workstation Scheduling Tests
…

Reference Schedule S4:
Identical to schedule S1, except that the Exception_Schedule property is empty.

[Move 13.10.4.7, 13.10.4.8, 13.10.4.9 and 13.10.4.10 from 135.1 2025 to BTL Specified Tests]

13.10.4.7 Modify an Exception_Schedule by Adding a BACnetSpecialEvent with Period of Choice calendarEntry of choice Date

Reason for Change: Resolve issues with adding a schedule to a full reference schedule. Using a new empty reference schedule S4.

Purpose: Demonstrate that the IUT can accept user input and modify the Exception_Schedule by adding a BACnetSpecialEvent of the schedule with the calendarEntry of type Date.
Test Concept: The tester will sit outside on a cloudless night when the moon is not a new moon. The tester waits until the moon is halfway through its traversal of the sky and then verifies the height above the horizon.

ConfigurationRequirements: A reference device contains reference objects ~~S1~~*S4*, C1, and C2.

Test Steps:

1. MAKE(the IUT add an entire BACnetSpecialEvent with period of choice calendarEntry of choice Date to the Exception_Schedule of ~~S1~~*S4*)
2. CHECK(Did the IUT write the modified Exception_Schedule correctly?)

Notes to Tester: For example, try different Date representations that leave one or more of each of the following fields unspecified: year, month, dayOfMonth, dayOfWeek. If the IUT supports devices with Protocol_Revision >= 4, try using the special values for month 13 (all odd) and 14 (all even) and the special value of 32 (last day) for the dayOfMonth.

13.10.4.8 Modify an Exception_Schedule by Adding a BACnetSpecialEvent with Period of Choice calendarEntry of Choice DateRange

Reason for Change: Resolve issues with adding a schedule to a full reference schedule. Using a new empty reference schedule S4.

Purpose: Demonstrate that the IUT can accept user input and modify the Exception_Schedule by adding a BACnetSpecialEvent of the schedule with the calendarEntry of type DateRange.

Configuration Requirements: A reference device contains reference objects ~~S1~~*S4*, C1, and C2.

Test Steps:

1. MAKE(the IUT add an entire BACnetSpecialEvent with period of choice calendarEntry of choice DateRange to the Exception_Schedule of ~~S1~~*S4*)
2. CHECK(Did the IUT write the modified Exception_Schedule correctly?)

Notes to Tester: It is not required that the IUT be capable of creating DateRanges with wildcard fields with the exception of unspecified dates (all fields equal to 255 in either one or both of the date fields of the BACnetDateRange).

13.10.4.9 Modify an Exception_Schedule by Adding a BACnetSpecialEvent with Period of Choice calendarEntry of Choice WeekNDay

Reason for Change: Resolve issues with adding a schedule to a full reference schedule. Using a new empty reference schedule S4.

Purpose: Demonstrate that the IUT can accept user input and modify the Exception_Schedule by adding a

© 2026 by BACnet International. All rights reserved.

BACnetSpecialEvent of the schedule with the calendarEntry of type WeekNDay.
Test Concept: For different WeekNDay values, add a BACnetCalendarEntry to the Date_List property of an Exception_Schedule, and verify the IUT writes the modified Date_List correctly to the reference device.

Configuration Requirements: A reference device contains reference objects ~~S1~~*S4*, C1, and C2. Choose the value set to use from the table below based on the protocol-revision for the IUT.

		WeekNDay = <Month><weekOfMonth><dayOfWeek>		
Value Set	Protocol_Revision	Month	weekOfMonth	dayOfWeek
1	PR < 4	0 to 12 and FF	1 to 6 and FF	1 to 7 and FF
2	PR >= 4	0 to 12 and FF and 13 and 14	1 to 6 and FF	1 to 7 and FF
3	PR >= 18	0 to 12 and FF and 13 and 14	1 to 9 and FF	1 to 7 and FF

Test Steps:

REPEAT WND = (WeekNDay Value Set from table) DO {
1. MAKE(the IUT add an entire BACnetSpecialEvent with period of choice calendarEntry of choice WeekNDay with value of WND to the Exception_Schedule of ~~S1~~*S4*)
2. CHECK(Did the IUT write the modified Exception_Schedule correctly?)

}

13.10.4.10 Modify an Exception_Schedule by Adding a BACnetSpecialEvent with Period of Choice calendarReference
Reason for Change: Resolve issues with adding a schedule to a full reference schedule. Using a new empty reference schedule S4.

Purpose: Demonstrate that the IUT can accept user input and modify the Exception_Schedule by adding a BACnetSpecialEvent of the schedule with the calendarEntry of type calendarReference.

Configuration Requirements: A reference device contains reference objects ~~S1~~*S4*, C1, and C2.

Test Steps:

1. MAKE(the IUT add an entire BACnetSpecialEvent with period of choice calendarReference to the Exception_Schedule of ~~S1~~*S4*)
2. CHECK(Did the IUT write the modified Exception_Schedule correctly?)

Notes to Tester: For example, add a BACnetSpecialEvent that is another CalendarReference to C2.

© 2026 by BACnet International. All rights reserved.

BTL-26.1 cr-imp1-2: Update DM-DCC and DM-RD B-side Testing [BTLWG-1726, CR-0582]

Overview:

If the IUT does not support initiation yet is required to support DM-DCC-B, many of the DCC tests are skipped thus other aspects of DCC are not tested. The tests are almost the same with password or no password, indefinite or finite, DISABLE or DISABLE_INITIATION and restored using DDC or RD. DM-RD-B is only tested with finite duration so should be a test conditionality not a separate checklist item.

Changes:

Checklist Changes

Device Management - Device Communication Control - B		
	R	Base Requirements
	C[1]	Supports receiving a DeviceCommunicationControl service request with a password
	C[1]	Supports receiving a DeviceCommunicationControl service request with no password
	R	Supports receiving a DeviceCommunicationControl service request with a finite duration
	O	Supports receiving a DeviceCommunicationControl service request with an indefinite duration
	~~O~~	~~Supports DM-RD-B~~
	~~C[4]~~	~~Supports receiving a DeviceCommunicationControl service request specifying DISABLE_INITIATION.~~
[1] At least one of these options is required in order to claim conformance to this BIBB. ~~[4] Required if device implements protocol revision 4 or higher.~~		

Test Plan Changes

8.14 Device Management - Device Communication Control - B

8.14.1 Base Requirements

~~**135.1-2025 - 9.24.1.11 - Ensure that DISABLE Option is not Supported by IUT Claiming PR >= 20**~~		
	~~**Test Conditionality**~~	~~If the IUT claims Protocol_Revision < 20, this test shall be skipped.~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	
BTL - 9.24.1.X1 - DeviceCommunicationControl Enable-Disable Test		
	Test Conditionality	*Must be executed.*
	Test Directives	
	Testing Hints	

8.14.2 Supports Receiving a DeviceCommunicationControl Service Request with a Password

The IUT requires, or can be made to require, a valid password parameter in a DeviceCommunicationControl service request.

~~**135.1-2025*BTL***~~ **- 9.24.2.1 - Invalid Password**		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
~~**135.1-2025*BTL***~~ **- 9.24.2.2 - Missing Password**		
	Test Conditionality	Must be executed.

© 2026 by BACnet International. All rights reserved.

	Test Directives	
	Testing Hints	

8.14.3 Supports Receiving a DeviceCommunicationControl Service Request with no Password

The IUT does not require, or can be made to not require, a password parameter in a DeviceCommunicationControl service request.

BTL - 9.24.1.X5 - DCC Password Not Required		
	Test Conditionality	*Must be Executed.*
	Test Directives	
	Testing Hints	
~~**135.1-2025 - 9.24.1.8 - Finite Time Duration, Disable Initiation**~~		
	~~**Test Conditionality**~~	~~If the IUT does not initiate any services, other than an I-Am in response to a Who-Is, then this test case shall be skipped.~~
	~~**Test Directives**~~	~~The service request shall not contain a password.~~
	~~**Testing Hints**~~	
~~**135.1-2025 - 9.24.1.6 - Indefinite Time Duration, Disable Initiation, Restored by DeviceCommunicationControl**~~		
	~~**Test Conditionality**~~	~~If the IUT does not initiate any services, other than an I-Am in response to a Who-Is, then this test case shall be skipped.~~
	~~**Test Directives**~~	~~The service request shall not contain a password.~~
	~~**Testing Hints**~~	

8.14.4 Supports Receiving a DeviceCommunicationControl Service Request with a Finite Duration

The IUT will accept, or can be made to accept, a DeviceCommunicationControl Service request with a Time Duration parameter.

~~**135.1-2025 - 9.24.1.3 - Finite Time Duration**~~		
	~~**Test Conditionality**~~	~~If the IUT claims Protocol_Revision >= 20, this test shall be skipped.~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	
~~**135.1-2025 - 9.24.1.4 - Finite Time Duration Restored by DeviceCommunicationControl**~~		
	~~**Test Conditionality**~~	~~If the IUT claims Protocol_Revision >= 20, this test shall be skipped.~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	
~~**135.1-2025 - 9.24.1.8 - Finite Time Duration, Disable Initiation**~~		
	~~**Test Conditionality**~~	~~Must be executed.~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	
BTL - 9.24.1.X2 - DCC Restored with Time Expired		
	Test Conditionality	*Must be executed.*
	Test Directives	*Execute this test once with DISABLE_INITIATION and if the IUT claims Protocol_Revision < 20, once with DISABLE. The Time Duration parameter shall be present in the DeviceCommunicationControl Service.*
	Testing Hints	
BTL - 9.24.1.X3 - DCC Restored by DeviceCommunicationControl		
	Test Conditionality	*Must be executed.*
	Test Directives	*Execute this test once with DISABLE_INITIATION and if the IUT claims Protocol_Revision < 20, once with DISABLE. The Time Duration parameter shall be present in the DeviceCommunicationControl Service.*
	Testing Hints	
BTL - 9.24.1.X4 - DCC Restored by ReinitializeDevice		

© 2026 by BACnet International. All rights reserved.

	Test Conditionality	*Must be executed if DM-RD-B is claimed.*
	Test Directives	*Execute this test once with DISABLE_INITIATION and if the IUT claims Protocol_Revision < 20, once with DISABLE. The Time Duration parameter shall be present in the DeviceCommunicationControl Service.*
	Testing Hints	

8.14.5 Supports Receiving a DeviceCommunicationControl Service Request with an Indefinite Duration

The IUT will accept, or can be made to accept, a DeviceCommunicationControl Service request with no Time Duration parameter.

~~**135.1-2025 - 9.24.1.1 - Indefinite Time Duration Restored by DeviceCommunicationControl**~~		
	~~**Test Conditionality**~~	~~If the IUT claims Protocol_Revision >= 20, this test shall be skipped.~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	
~~**135.1-2025 - 9.24.1.6 - Indefinite Time Duration, Disable-Initiation, Restored by DeviceCommunicationControl**~~		
	~~**Test Conditionality**~~	~~Must be executed.~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	
BTL - 9.24.1.X3 - DCC Restored by DeviceCommunicationControl		
	Test Conditionality	*Must be executed.*
	Test Directives	*Execute this test once with DISABLE_INITIATION and if the IUT claims Protocol_Revision < 20, once with DISABLE. The Time Duration parameter shall be absent in the DeviceCommunicationControl Service.*
	Testing Hints	
BTL - 9.24.1.X4 - DCC Restored by ReinitializeDevice		
	Test Conditionality	*Must be executed if DM-RD-B is claimed.*
	Test Directives	*Execute this test once with DISABLE_INITIATION and if the IUT claims Protocol_Revision < 20, once with DISABLE. The Time Duration parameter shall be absent in the DeviceCommunicationControl Service.*
	Testing Hints	

~~8.14.6Supports DM-RD-B~~

~~The IUT also supports the DM-RD-B BIBB.~~

~~**135.1-2025 - 9.24.1.2 - Indefinite Time Duration Restored by ReinitializeDevice**~~		
	~~**Test Conditionality**~~	~~If the IUT claims Protocol_Revision >= 20, this test shall be skipped.~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	
~~**135.1-2025 - 9.24.1.5 - Finite Time Duration Restored by ReinitializeDevice**~~		
	~~**Test Conditionality**~~	~~If the IUT claims Protocol_Revision >= 20, this test shall be skipped.~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	
~~**135.1-2025 - 9.24.1.7 - Indefinite Time Duration, Disable-Initiation, Restored by ReinitializeDevice**~~		
	~~**Test Conditionality**~~	~~If the IUT supports indefinite Time Duration and is capable of initiating service requests other than I-Am, this test must be executed.~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	
~~**135.1-2025 - 9.24.1.12 - Disable of Service Initiation Restored by ReinitializeDevice**~~		

© 2026 by BACnet International. All rights reserved.

	Test Conditionality	If the IUT does not initate any services other than an I-Am in response to a who-Is, then this test shall be skipped.
	Test Directives	
	Testing Hints	
135.1-2025 - 9.24.2.3 - Restore by ReinitializeDevice with Invalid 'Reinitialized State of Device'		
	Test Conditionality	If the IUT claims Protocol_Revision >= 20, this test shall be skipped.
	Test Directives	
	Testing Hints	

8.14.7 Supports Receiving a DeviceCommunicationControl Service Request Specifying DISABLE_INITIATION

The IUT will accept the value of DISABLE_INITIATION in the enable-disable parameter of the service request.

135.1-2025 - 9.24.1.9 - Disable of Service Initiation Restored by Time Duration		
	Test Conditionality	If the IUT cannot be made to initiate any service, this test shall be skipped.
	Test Directives	
	Testing Hints	
135.1-2025 - 9.24.1.10 - Disable of Service Initiation Restored by DeviceCommunicationControl		
	Test Conditionality	If the IUT cannot be made to initiate any service, this test shall be skipped.
	Test Directives	
	Testing Hints	

Specified Test Changes

9.24.1.1 Deleted Clause

This test has been removed.

9.24.1.1 Indefinite Time Duration Restored by DeviceCommunicationControl

Purpose: To verify the correct execution of the DeviceCommunicationControl request service procedure when indefinite time duration is specified and communication is restored using the DeviceCommunicationControl service.

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request,
   'Enable/Disable' = DISABLE,
   'Password' = (any appropriate password as described in the Test Concept)
2. RECEIVE BACnet-SimpleACK-PDU
3. WAIT **Internal Processing Fail Time**
4. WHILE (an arbitrary time > **Internal Processing Fail Time** selected by the tester has not expired) DO {
   TRANSMIT ReadProperty-Request,
   'Object Identifier' = (Device, X),
   'Property Identifier' = (any required non-array property of the Device object)
   TRANSMIT
   DESTINATION = LOCAL-BROADCAST,
   Who-Is-Request
   WAIT (1 second) poll delay
   }
5. CHECK (Verify that the IUT has not transmitted any APDUs since the acknowledgment in step 2)
6. TRANSMIT DeviceCommunicationControl-Request,
   'Enable/Disable' = ENABLE,
   'Password' = (any appropriate password as described in the Configuration Requirements)

© 2026 by BACnet International. All rights reserved.

7. RECEIVE BACnet-SimpleACK-PDU
8. VERIFY (Device, X), (any required non-array property) = (any valid value)

9.24.1.2 Deleted Clause

This test has been removed.

9.24.1.2 Indefinite Time Duration Restored by ReinitializeDevice

Purpose: To verify the correct execution of the DeviceCommunicationControl request service procedure when indefinite time duration is specified and communication is restored using the ReinitializeDevice service.

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request,
   'Enable/Disable' = DISABLE,
   'Password' = (any appropriate password as described in the Test Concept)
2. RECEIVE BACnet-SimpleACK-PDU
3. WAIT **Internal Processing Fail Time**
4. WHILE (an arbitrary time > **Internal Processing Fail Time** selected by the tester has not expired) DO {
   TRANSMIT ReadProperty-Request,
   'Object Identifier' = (Device, X),
   'Property Identifier' = (any required non-array property of the Device object)
   TRANSMIT
   DESTINATION = LOCAL-BROADCAST,
   Who-Is-Request
   WAIT (1 second) poll delay
   }
5. CHECK (Verify that the IUT has not transmitted any APDUs since the acknowledgment in step 2)
6. TRANSMIT ReinitializeDevice-Request,
   'Reinitialized State of Device' = WARMSTART,
   'Password' = (any appropriate password as described in the Test Concept)
7. RECEIVE BACnet-SimpleACK-PDU
8. CHECK (Did the IUT perform a WARMSTART reboot?)
9. VERIFY (Device, X), (any required non-array property) = (any valid value)

9.24.1.3 Deleted Clause

This test has been removed.

9.24.1.3 Finite Time Duration

Purpose: To verify the correct execution of the DeviceCommunicationControl request service procedure when finite time duration is specified.

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request,
   'Time Duration' = (a value T > 1, in minutes, selected by the tester),
   'Enable/Disable' = DISABLE,
   'Password' = (any appropriate password as described in the Test Concept)
2. RECEIVE BACnet-SimpleACK-PDU
3. WAIT **Internal Processing Fail Time**
4. WHILE (Time Duration T in step 1 has not expired) DO {
   TRANSMIT ReadProperty-Request,
   'Object Identifier' = (Device, X),
   'Property Identifier' = (any required non-array property of the Device object)
   TRANSMIT
   DESTINATION = LOCAL-BROADCAST,
   Who-Is-Request
   WAIT (1 second) poll delay
   }

© 2026 by BACnet International. All rights reserved.

5. CHECK (Verify that the IUT did not transmit any APDUs between the acknowledgment in step 2 and expiration of timer T)
6. VERIFY (Device, X), (any required non-array property) = (any valid value)

9.24.1.4 Deleted Clause

This test has been removed.

9.24.1.4 Finite Time Duration Restored by DeviceCommunicationControl

Purpose: To verify the correct execution of the DeviceCommunicationControl request service procedure when a finite time duration is specified and communication is restored using the DeviceCommunicationControl service.

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request,
   'Time Duration' = (a value T > 1, in minutes, selected by the tester),
   'Enable/Disable' = DISABLE,
   'Password' = (any appropriate password as described in the Test Concept)
2. RECEIVE BACnet-SimpleACK-PDU
3. WAIT **Internal Processing Fail Time**
4. TRANSMIT ReadProperty-Request,
   'Object Identifier' = (Device, X),
   'Property Identifier' = (any required non-array property of the Device object)
5. WAIT (an arbitrary time > **Internal Processing Fail Time** selected by the tester, and < T as specified in the DeviceCommunicationControl-Request)
6. CHECK (Verify that the IUT has not transmitted any APDUs since the acknowledgment in step 2)
7. TRANSMIT DeviceCommunicationControl-Request,
   'Enable/Disable' = ENABLE,
   'Password' = (any appropriate password as described in the Configuration Requirements)
8. RECEIVE BACnet-SimpleACK-PDU
9. VERIFY ((Device, X), (any required non-array property) = (the value for this property as described in the Configuration Requirements))

9.24.1.5 Deleted Clause

This test has been removed.

9.24.1.5 Finite Time Duration Restored by ReinitializeDevice

Purpose: To verify the correct execution of the DeviceCommunicationControl request service procedure when a finite time duration is specified and communication is restored using the ReinitializeDevice service.

Test Steps:

1. READ Y = (Device, X), Object_Name
2. TRANSMIT DeviceCommunicationControl-Request,
   'Time Duration' =(a value T > 1, in minutes, selected by the tester)
   'Enable/Disable' = DISABLE,
   'Password' = (any appropriate password as described in the Test Concept)
3. RECEIVE BACnet-SimpleACK-PDU
4. WAIT **Internal Processing Fail Time**
5. TRANSMIT ReadProperty-Request,
   'Object Identifier' = (Device, X),
   'Property Identifier' = (any required non-array property of the Device object)
6. WAIT (an arbitrary time > **Internal Processing Fail Time** selected by the tester, and < T as specified in the DeviceCommunicationControl-Request)
7. CHECK (Verify that the IUT has not transmitted any APDUs since the acknowledgment in step 2.)
8. TRANSMIT ReinitializeDevice-Request,
   'Reinitialize State of Device' = WARMSTART,
   'Password' = (any appropriate password as described in the Test Concept)
9. RECEIVE BACnet-SimpleACK-PDU

© 2026 by BACnet International. All rights reserved.

10. CHECK (Did the IUT perform a WARMSTART reboot?)
11. VERIFY (Device, X), Object_Name = Y

9.24.1.6 Deleted Clause

This test has been removed.

9.24.1.6 Indefinite Time Duration, Disable-Initiation, Restored by DeviceCommunicationControl

Purpose: This test case verifies the correct execution of the DeviceCommunicationControl request service procedure when an indefinite time duration is specified, only initiation is disabled, and communication is restored using the DeviceCommunicationControl service. If the IUT does not initiate any services other than an I-Am in response to a Who-Is, then this test case shall be skipped.

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request,
   'Enable/Disable' = DISABLE-INITIATION,
   'Password' = (any appropriate password as described in the Test Concept)
2. RECEIVE BACnet-SimpleACK-PDU
3. WAIT **Internal Processing Fail Time**
4. MAKE (do something that would normally cause the IUT to initiate a message)
5. WAIT (an arbitrary time > **Internal Processing Fail Time** selected by the tester)
6. CHECK (Verify that the IUT has not transmitted any APDUs since the acknowledgment in step 2.)
7. TRANSMIT ReadProperty-Request,
   'Object Identifier' = (Device, X),
   'Property Identifier' = (any required non-array property of the Device object)
8. RECEIVE BACnet-ComplexACK-PDU,
   'Object Identifier' = (Device, X),
   'Property Identifier' = (the 'Property Identifier' specified in step 7),
   'Property Value' = (any valid value for the property)
9. TRANSMIT DeviceCommunicationControl-Request,
   'Enable/Disable' = ENABLE,
   'Password' = (any appropriate password as described in the Test Concept)
10. RECEIVE BACnet-SimpleACK-PDU
11. MAKE (do something to cause the IUT to initiate an APDU)
12. WAIT **Internal Processing Fail Time**
13. CHECK (Verify that the IUT initiated an APDU)

9.24.1.7 Deleted Clause

This test has been removed.

9.24.1.7 Indefinite Time Duration, Disable-Initiation, Restored by ReinitializeDevice

Purpose: This test case verifies the correct execution of the DeviceCommunicationControl request service procedure when an indefinite time duration is specified, only initiation is disabled, and communication is restored using the ReinitializeDevice service. If the IUT does not initiate any services other than an I-Am in response to a Who-Is, then this test case shall be skipped.

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request,
   'Enable/Disable' = DISABLE-INITIATION,
   'Password' = (any appropriate password as described in the Test Concept)
2. RECEIVE BACnet-SimpleACK-PDU
3. WAIT **Internal Processing Fail Time**
4. MAKE (do something that would normally cause the IUT to initiate a message)
5. WAIT (an arbitrary time > **Internal Processing Fail Time** selected by the tester)
6. CHECK (Verify that the IUT has not transmitted any APDUs since the acknowledgment in step 2.)
7. TRANSMIT ReadProperty-Request,
   'Object Identifier' = (Device, X),

© 2026 by BACnet International. All rights reserved.

'Property Identifier' = (any required non-array property of the Device object)
8. RECEIVE BACnet-ComplexACK-PDU,
'Object Identifier' = (Device, X),
'Property Identifier' = (the 'Property Identifier' specified in step 7),
'Property Value' = (any valid value for the property)
9. TRANSMIT ReinitializeDevice-Request,
'Reinitialized State of Device' = WARMSTART,
'Password' = (any appropriate password as described in the Test Concept)
10. RECEIVE BACnet-SimpleACK-PDU
11. CHECK (Did the IUT perform a WARMSTART reboot?)
12. MAKE (do something to cause the IUT to initiate an APDU)
13. WAIT **Internal Processing Fail Time**
14. CHECK (Verify that the IUT initiated an APDU)

9.24.1.8 Deleted Clause

This test has been removed.

9.24.1.8 Finite Time Duration, Disable Initiation

Purpose: This test case verifies the correct execution of the DeviceCommunicationControl request service procedure when finite time duration is specified and only initiation is disabled. If the IUT does not initiate any services, other than an I-Am in response to a Who-Is, then this test case shall be skipped.

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request,
'Time Duration' = (a value T > 1, in minutes, selected by the tester).
'Enable/Disable' = DISABLE_INITIATION,
'Password' = (any appropriate password as described in the Test Concept)
2. RECEIVE BACnet-SimpleACK-PDU
3. WAIT **Internal Processing Fail Time**
4. MAKE (do something that would normally cause the IUT to initiate a message)
5. WAIT (an arbitrary time > **Internal Processing Fail Time** selected by the tester)
6. CHECK (Verify that the IUT has not transmitted any APDUs since the acknowledgment in step 2.)
7. TRANSMIT ReadProperty-Request,
'Object Identifier' = (Device, X),
'Property Identifier' = (any required non-array property of the Device object)
8. RECEIVE BACnet-ComplexACK-PDU,
'Object Identifier' = (Device, X),
'Property Identifier' = (the 'Property Identifier' specified in step 7),
'Property Value' = (any valid value for the property)
9. WAIT (T)
10. MAKE (do something to cause the IUT to initiate aan APDU)
11. WAIT **Internal Processing Fail Time**
12. CHECK (Verify that the IUT initiated an APDU)

9.24.1.9 Deleted Clause

This test has been removed.

9.24.1.9 Disable of Service Initiation Restored by Time Duration

Purpose: To verify the correct execution of the DeviceCommunicationControl service when DISABLE_INITIATION is requested with a finite time duration. Communication is restored when the DeviceCommunicationControl 'Time Duration' parameter expires.

Configuration Requirements: The IUT shall be configured to initiate client requests.

Test Steps:

1. MAKE (a condition that would normally cause the IUT to initiate requests)

© 2026 by BACnet International. All rights reserved.

2. CHECK (that the IUT is initiating requests)
3. TRANSMIT DeviceCommunicationControl-Request,
'Time Duration' = (a value T > 1, in minutes, selected by the tester),
'Enable/Disable' = DISABLE_INITIATION,
'Password' = (any appropriate password if required)
4. RECEIVE BACnet-SimpleACK-PDU
5. MAKE (a condition that would normally cause the IUT to initiate requests)
6. CHECK (that the IUT has stopped initiating requests)
7. VERIFY (any supported property) = (any valid value)
8. TRANSMIT Who-Is-Request
9. RECEIVE I-Am-Request
10. WAIT (time T)
11. MAKE (a condition that would normally cause the IUT to initiate requests)
12. CHECK (that the IUT is initiating requests)

Notes to Tester: Steps 2 through 9 must be executed within time T.

9.24.1.10 Deleted Clause

This test has been removed.

9.24.1.10 Disable of Service Initiation Restored by DeviceCommunicationControl

Purpose: To verify the correct execution of the DeviceCommunicationControl service when DISABLE_INITIATION is requested for a finite time duration. Communication is restored using the DeviceCommunicationControl service.

Configuration Requirements: The IUT shall be configured to initiate client requests.

Test Steps:

1. MAKE (a condition that would normally cause the IUT to initiate requests)
2. CHECK (that the IUT is initiating requests)
3. TRANSMIT DeviceCommunicationControl-Request,
'Time Duration' = (a value in minutes > time required to execute all test steps),
'Enable/Disable' = DISABLE_INITIATION,
'Password' = (any appropriate password if required)
4. RECEIVE BACnet-SimpleACK-PDU
5. MAKE (a condition that would normally cause IUT to initiate requests)
6. CHECK (that the IUT has stopped initiating requests)
7. VERIFY (any supported property) = (any valid value)
8. TRANSMIT Who-Is-Request
9. RECEIVE I-Am-Request
10. TRANSMIT DeviceCommunicationControl-Request,
'Enable/Disable' = ENABLE,
'Password' = (any appropriate password if required)
11. MAKE (a condition that would normally cause the IUT to initiate requests)
12. CHECK (that the IUT is initiating requests)

9.24.1.11 Deleted Clause

This test has been removed.

9.24.1.11 Ensure that DISABLE Option is not Supported by IUT Claiming PR >= 20

Purpose: To verify that IUT claiming Protocol Revision (PR) greater than or equal to 20, does not accept 'Enable/Disable' parameter equal to DISABLE in the DeviceCommunicationControl request.

Test Concept: Send DeviceCommunicationControl request with 'Enable/Disable' parameter equal to DISABLE to IUT. Then IUT is verified that it correctly responds with a Result( ).

Configuration Requirements: If the IUT does not support an internal clock this test shall be tested with indefinite time duration.

© 2026 by BACnet International. All rights reserved.

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request
   'Time Duration' = (a value T>1, in minutes) | (no value)
   'Enable/Disable'= DISABLE,
   'Password' = Any appropriate password if required
2. RECEIVE BACnet-Error-PDU
   Error Class = SERVICES,
   Error Code = SERVICE_REQUEST_DENIED
3. VERIFY (Device, X), System_Status = (any valid value)

9.24.1.12 Deleted Clause

This test has been removed.

9.24.1.12 Disable of Service Initiation Restored by ReinitializeDevice

Purpose: To verify the correct execution of the DeviceCommunicationControl service when DISABLE_INITIATION is requested with a finite time duration. Communication is restored using the ReinitializeDevice service.

Configuration Requirements: The IUT shall be configured to initiate client requests.

Test Steps:

1. MAKE (a condition that would normally cause the IUT to initiate requests)
2. CHECK (that the IUT is initiating requests)
3. TRANSMIT DeviceCommunicationControl-Request,
   'Time Duration' = (a value in minutes > time required to execute all test steps),
   'Enable/Disable' = DISABLE_INITIATION,
   'Password' = (any appropriate password if required)
4. RECEIVE BACnet-SimpleACK-PDU
5. MAKE (a condition that would normally cause IUT to initiate requests)
6. CHECK (that the IUT has stopped initiating requests)
7. VERIFY (any supported property) = (any valid value)
8. TRANSMIT Who-Is-Request
9. RECEIVE I-Am-Request
10. TRANSMIT ReinitializeDevice-Request,
    'Reinitialized State of Device' = WARMSTART,
    'Password' = (any appropriate password)
11. RECEIVE BACnet-SimpleACK-PDU
12. CHECK (Did the IUT perform a WARMSTART reboot?)
13. MAKE (a condition that would normally cause the IUT to initiate requests)
14. CHECK (that the IUT is initiating requests)

9.24.2.3 Deleted Clause

This test has been removed.

9.24.2.3 Restore by ReinitializeDevice with Invalid 'Reinitialized State of Device'

Purpose: To verify that the communications are not restored when a ReinitializeDevice request is received containing one of the backup or restore related values for service parameter 'Reinitialized State of Device'.

Test Concept: Disable the IUT's communications for a period time, T, which shall be longer than it will take to complete the test. Verify that, while communications are disabled, the IUT correctly responds with a Result( ) when it receives a ReinitializeDevice request containing a backup or restore related values.

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request,
   'Enable/Disable' = DISABLE

© 2026 by BACnet International. All rights reserved.

~~'Password' = (any appropriate password),~~
~~'Time Duration' = (a value T >= 1, in minutes) | (no value)~~
~~2. RECEIVE BACnet-SimpleACK-PDU~~
~~3. WAIT **Internal Processing Fail Time**~~
~~4. TRANSMIT ReinitializeDevice-Request,~~
~~'Reinitialized State of Device' = STARTBACKUP | ENDBACKUP | STARTRESTORE | ENDRESTORE | ABORTRESTORE,~~
~~'Password' = (any appropriate password)~~
~~5. IF (Protocol_Revision is present AND Protocol_Revision >= 7) THEN~~
~~IF (Device supports DM-BR-B) THEN~~
~~RECEIVE BACnet-Error-PDU,~~
~~Error Class = SERVICES,~~
~~Error Code = COMMUNICATION_DISABLED~~
~~ELSE~~
~~RECEIVE BACnet-Error-PDU,~~
~~'Error Class' = SERVICES,~~
~~'Error Code' = COMMUNICATION_DISABLED | OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED~~
~~ELSE~~
~~CHECK(that the IUT responded with BACnet-Error-PDU with an Error Class of SERVICES and any appropriate Error Code or that the IUT did not respond at all)~~
~~6. TRANSMIT DeviceCommunicationControl-Request,~~
~~'Enable/Disable' = ENABLE~~
~~'Password' = (any appropriate password),~~
~~7. RECEIVE BACnet-SimpleACK-PDU~~

[Add new test to BTL Specified Tests]

9.24.1.X1 DCC Enable-Disable Test

Reason For Change: If the IUT does not support initiation, many of the tests are skipped so other aspects of DCC are not tested. This test validates simple and more complex IUTs.

Purpose: To verify the IUT correctly executes the actions of DISABLE, DISABLE_INITIATION, and ENABLE.

Test Concept: Send DISABLE and ENABLE and verify the IUT accepts the commands. Send DISABLE_INITIATION and verify read requests, Who-Is/I-Am, and Who-Has/I-Have function and the IUT does not initiate any other requests.

Notes to Tester: Execute this test with the Time Duration parameter present. If the IUT does not require or can be configured to not require a Password, execute this test with any string in the Password parameter. If the IUT requires a Password, execute this test with a valid password in the Password parameter.

Test Steps:

-- test DISABLE
1. IF (Protocol_Revision < 20) THEN {
2. TRANSMIT DeviceCommunicationControl-Request,
'Time Duration' = (any valid value long enough to execute this test),
'Enable/Disable' = DISABLE,
'Password' = (an appropriate password)
3. RECEIVE BACnet-SimpleACK-PDU
4. TRANSMIT DeviceCommunicationControl-Request,
'Enable/Disable' = ENABLE,
'Password' = (an appropriate password)
5. RECEIVE BACnet-SimpleACK-PDU
6. CHECK (Verify that the IUT has not transmitted any APDUs since the acknowledgment in step 2)
}
ELSE {
7. TRANSMIT DeviceCommunicationControl-Request,
'Time Duration' = (any valid value long enough to execute this test),
'Enable/Disable' = DISABLE,

© 2026 by BACnet International. All rights reserved.

```
                'Password' =            (an appropriate password)
8.          RECEIVE BACnet-Error-PDU
                Error Class = SERVICES,
                Error Code = SERVICE_REQUEST_DENIED
        }

 -- test DISABLE_INITIATION
9.    TRANSMIT DeviceCommunicationControl-Request,
            'Time Duration' =       (any valid value long enough to execute this test),
            'Enable/Disable' =      DISABLE_INITIATION,
            'Password' =            (an appropriate password)
10.   RECEIVE BACnet-SimpleACK-PDU
 -- test the IUT still responds to requests
11.   VERIFY (any supported property) = (any valid value)
-- test I-Am
12.   TRANSMIT Who-Is-Request
13.   RECEIVE I-Am-Request
14.   IF (Protocol_Revision >= 30) THEN {
            -- test I-Have
15.         TRANSMIT Who-Has-Request
                'Object Name' = (O1, any object in the IUT)
16.         RECEIVE I-Have-Request
                'Device Identifier' = (the IUT's Device object),
                'Object Identifier' = O1,
                'Object Name' = (O1's object name)
        }
17.   IF (the IUT can initiate a message other than I-Am in response to Who-Is,
                I-Have in response to Who-Has, or
                ConfirmedAuditNotification and UnconfirmedAuditNotification requests) THEN {
            -- test the IUT does not initiate any requests other than specified
18.         MAKE (do something that would normally cause the IUT to initiate a message)
19.         WAIT (an arbitrary time > Internal Processing Fail Time selected by the tester)
20.         CHECK (Verify that the IUT has not transmitted any APDUs)
        }
21.   TRANSMIT DeviceCommunicationControl-Request,
            'Enable/Disable' =      ENABLE,
            'Password' =            (an appropriate password)
22.   RECEIVE BACnet-SimpleACK-PDU
```

9.24.1.X2 DCC Restored with Time Expired

Reason For Change: Replaces 9.24.1.3, 9.24.1.8, 9.24.1.9.

Purpose: To verify the correct execution of the DeviceCommunicationControl request service procedure when finite time duration is specified.

Test Steps:

```
1.    TRANSMIT DeviceCommunicationControl-Request,
            'Time Duration' =       (a value T > 1, in minutes, selected by the tester),
            'Enable/Disable' =      (parameter configured for this test),
            'Password' =            (absent or an appropriate password)
2.    RECEIVE BACnet-SimpleACK-PDU
3.    WAIT (T)
4.    VERIFY (any supported property) = (any valid value)
```

9.24.1.X3 DCC Restored by DeviceCommunicationControl

Reason For Change: Replaces 9.24.1.1, 9.24.1.4, 9.24.1.6, 9.24.1.10.

© 2026 by BACnet International. All rights reserved.

Purpose: To verify the correct execution of the DeviceCommunicationControl request service procedure communication is restored using the DeviceCommunicationControl service.

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request,
   'Time Duration' = (absent or a value T > 1, in minutes, selected by the tester),
   'Enable/Disable' = (parameter configured for this test),
   'Password' = (absent or an appropriate password)
2. RECEIVE BACnet-SimpleACK-PDU
3. TRANSMIT DeviceCommunicationControl-Request,
   'Enable/Disable' = ENABLE,
   'Password' = (absent or an appropriate password)
4. RECEIVE BACnet-SimpleACK-PDU
5. VERIFY (any supported property) = (any valid value)

9.24.1.X4 DCC Restored by ReinitializeDevice

Reason For Change: Replaces 9.24.1.2, 9.24.1.5, 9.24.1.7, 9.24.1.12.

Purpose: To verify the correct execution of the DeviceCommunicationControl request service procedure communication is restored using the ReinitializeDevice service.

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request,
   'Time Duration' = (absent or a value T > 1, in minutes, selected by the tester),
   'Enable/Disable' = (parameter configured for this test),
   'Password' = (absent or an appropriate password)
2. RECEIVE BACnet-SimpleACK-PDU
3. TRANSMIT ReinitializeDevice-Request,
   'Reinitialize State of Device' = WARMSTART,
   'Password' = (absent or an appropriate password)
4. RECEIVE BACnet-SimpleACK-PDU
5. CHECK (Did the IUT perform a WARMSTART?)
6. VERIFY (any supported property) = (any valid value)

9.24.1.X5 DCC Password Not Required

Reason For Change: No test for this functionality. Clause 16.1.1.1.3 requires the Password to be ignored if not required.

Purpose: To verify the correct execution of the DeviceCommunicationControl request service procedure when a password is not required.

Test Steps:

1. TRANSMIT DeviceCommunicationControl-Request,
   'Time Duration' = (T, any valid value),
   'Enable/Disable' = DISABLE_INITIATION,
2. RECEIVE BACnet-SimpleACK-PDU
3. WAIT (T)
4. TRANSMIT DeviceCommunicationControl-Request,
   'Time Duration' = (T, any valid value),
   'Enable/Disable' = DISABLE_INITIATION,
   'Password' = (any valid arbitrary string)
5. RECEIVE BACnet-SimpleACK-PDU
6. WAIT (T)

© 2026 by BACnet International. All rights reserved.

BTL-26.1 cr-imp1-3: Initiate ReadRange Test Conditionality [BTLWG-1748, CR-0592]

Overview:

The Test Conditionality allows these tests to be skipped if the device also supports ReadRange by sequence number. The Test Conditionality also allows these tests to be executed if the IUT does not support ReadRange with no range or by position.

Changes:

Checklist Changes

Initiates ReadRange		
	R	Base Requirements
	C[1]	*Supports reading items with no specified range*
	C[1]	*Supports reading items by Position*
	C[1]	*Supports reading items by Time*
	C[1]	*Supports reading items by Sequence Number*
[1] *At least one of these options is required in order to claim conformance to this BIBB.*		

Test Plan Changes

4.15Initiates ReadRange

4.15.1 Base Requirements

~~Base requirements must be met by any IUT that initiates ReadRange.~~ *There are no base requirements for this service.*

~~**135.1-2025 - 8.21.1 - Reading Values with no Specified Range, or**~~ ~~**135.1-2025 - 8.21.3 - Reading a Range of Values by Position**~~		
	~~**Test Conditionality**~~	~~May be skipped if the IUT initiates ReadRange in another tested BIBB that uses ReadRange (eg T-ATR-A).~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	

[Add 4.15.2 to 4.15.5]

4.15.2 Supports reading items with no specified range

The IUT initiates ReadRange with no specified range.

135.1-2025 - 8.21.1 - Reading Values with no Specified Range		
	Test Conditionality	Must be executed if the IUT initiates a ReadRange request with no specified range on a property other than a log buffer.
	Test Directives	
	Testing Hints	
Verify Checklist		
	Test Conditionality	Must be executed if the IUT initiates ReadRange request with no specified range on a log buffer.
	Test Directives	Verify that the IUT claims support for at least one of these BIBBS. AE-ELV-A, T-V-A, T-ATR-A, T-AMVR-A, AR-V-A, or AR-AV-A.
	Testing Hints	

© 2026 by BACnet International. All rights reserved.

4.15.3 Supports reading items by Position

The IUT initiates ReadRange by Position.

135.1-2025 - 8.21.3 - Reading a Range of Values by Position		
	Test Conditionality	Must be executed if the IUT initiates a ReadRange request by Position on a property other than a log buffer.
	Test Directives	
	Testing Hints	
Verify Checklist		
	Test Conditionality	Must be executed if the IUT initiates ReadRange by Position on a log buffer.
	Test Directives	Verify that the IUT claims support for at least one of these BIBBS. AE-ELV-A, T-V-A, T-ATR-A, T-AMVR-A, AR-V-A, or AR-AV-A.
	Testing Hints	

4.15.4 Supports reading items by Time

The IUT initiates ReadRange by Time.

Verify Checklist		
	Test Conditionality	Must be executed if the IUT initiates ReadRange by Time.
	Test Directives	Verify that the IUT claims support for at least one of these BIBBS. AE-ELV-A, T-V-A, T-ATR-A, T-AMVR-A, AR-V-A, or AR-AV-A.
	Testing Hints	

4.15.5 Supports reading items by Sequence Number

The IUT initiates ReadRange by Sequence Number.

Verify Checklist		
	Test Conditionality	Must be executed if the IUT initiates ReadRange by Sequence Number.
	Test Directives	Verify that the IUT claims support for at least one of these BIBBS. AE-ELV-A, T-V-A, T-ATR-A, T-AMVR-A, AR-V-A, or AR-AV-A.
	Testing Hints	

Specified Test Changes

None

© 2026 by BACnet International. All rights reserved.

BTL-26.1 cr-imp1-4: Network Port - Unknown and Unsupported Commands [BTLWG-1767, CR-0596]

Overview:

In PR >= 24 devices, not supporting the NPO Command GENERATE_CSR_FILE and VALIDATE_CHANGES requires OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED to be returned. In PR < 24, either OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED or VALUE_OUT_OF_RANGE can be returned.

12.56.16 Command
…
If the Network Port object does not support a command written to this property, the device shall return a Result(-) with an 'Error Class' of PROPERTY and an 'Error Code' of OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED.
…
This enumerated value is extensible. Writing an unknown value to this property shall result in the return of a Result(-) with an 'Error Class' of PROPERTY and an 'Error Code' of VALUE_OUT_OF_RANGE.

Changes:

Checklist Changes

None

Test Plan Changes

[Change all references for 7.3.2.46.3.10.2 from 135.1 to BTL]
[Change all references for 7.3.2.46.3.10.4 from 135.1 to BTL]

Specified Test Changes

[Move tests from 135.1 into BTL Specified Tests and modify as shown]

7.3.2.46.3.10.2 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.

Test Steps:

1. TRANSMIT WriteProperty-Request,
   'Object Identifier' = (the Network Port object),
   'Property Identifier' = Command,
   'Property Value' = GENERATE_CSR_FILE
2. IF (Protocol_Revision >= 24) THEN
3. RECEIVE BACnet-Error-PDU
   'Error Class' = PROPERTY,
   'Error Code' = OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED
   ELSE
4. RECEIVE BACnet-Error-PDU
   'Error Class' = PROPERTY,

© 2026 by BACnet International. All rights reserved.

'Error Code' = OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED |
VALUE_OUT_OF_RANGE

7.3.2.46.3.10.4 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.

Test Steps:

1. TRANSMIT WriteProperty-Request,
   'Object Identifier' = (the Network Port object),
   'Property Identifier' = Command,
   'Property Value' = VALIDATE_CHANGES
2. IF (Protocol_Revision >= 24) THEN
3. RECEIVE BACnet-Error-PDU
   'Error Class' = PROPERTY,
   'Error Code' = OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED
   ELSE
4. RECEIVE BACnet-Error-PDU
   'Error Class' = PROPERTY,
   'Error Code' = OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED |
   VALUE_OUT_OF_RANGE

© 2026 by BACnet International. All rights reserved.

BTL-26.1 cr-imp1-5: Malformed BVLC Test for Nodes operating as a Hub Issues [BTLWG-1782, CR-0598]

Overview:

Add an unknown BVLC test with a DVA.
Fix Steps 10 and 1ll

Changes:

Checklist Changes

None

Test Plan Changes

[Change all references to 12.5.2.2.5 from 135.1-2025 to BTL]

Specified Test Changes

12.5.2.2.5 Malformed BVLC Test for Nodes operating as a Hub

Reason for Change: Added Unknown BVLC function check with DVA. Fixed addressing in Step 13(10) and 14(11).

Purpose: Verify the device NAKs malformed / unknown unicast BVLCs and ignores malformed / unknown broadcast BVLCs when operating as a Node with an active hub function.

Test Concept: With the IUT connected to the BACnet/SC network and operating as the active hub, send a sequence of malformed unicast and broadcast BVLCs to the IUT. Verify that the IUT responds as required to each message and does not process nor route the messages.

Configuration Requirements: The IUT is connected to the BACnet/SC network and is operating as the active hub. The TD is connected to the BACnet/SC network as a node.

Test Steps:

-- Unknown BVLC function – no DVA
1. TRANSMIT
   'BVLC Function' = (IV: an unknown 1-octet value),
   'Message ID' = (M1: any valid value),
   -- 'Originating Virtual Address' absent
   -- 'Destination Virtual Address' absent
   -- 'Destination Options' absent
   -- 'Data Options' absent
2. RECEIVE BVLC-Result,
   'Message ID' = M1,
   -- 'Originating Virtual Address' absent
   -- 'Destination Virtual Address' absent
   'Destination Options' = (absent or a valid list of options),
   -- 'Data Options' absent
   'Result for BVLC Function' = IV,-- the supplied unknown BVLC function from the request
   'Result Code' = X'01', -- NAK
   'Error Header Marker' = X'00', -- not a header option problem
   'Error Class' = COMMUNICATION,
   'Error Code' = BVLC_FUNCTION_UNKNOWN
3. CHECK(that the IUT did not process nor forward the request)

© 2026 by BACnet International. All rights reserved.

-- Unknown BVLC function – valid DVA
4. TRANSMIT
'BVLC Function' = (IV: an unknown 1-octet value),
'Message ID' = (M1: any valid value),
-- 'Originating Virtual Address' absent
'Destination Virtual Address' = (IUT's VMAC),
-- 'Destination Options' absent
-- 'Data Options' absent
5. RECEIVE BVLC-Result,
'Message ID' = M1,
'Originating Virtual Address' = (IUT's VMAC),
-- 'Destination Virtual Address' absent
'Destination Options' = (absent or a valid list of options),
-- 'Data Options' absent
'Result for BVLC Function' = IV,-- the supplied unknown BVLC function from the request
'Result Code' = X'01', -- NAK
'Error Header Marker' = X'00', -- not a header option problem
'Error Class' = COMMUNICATION,
'Error Code' = BVLC_FUNCTION_UNKNOWN
6. CHECK(that the IUT did not process nor forward the request)

-- Inclusion of an Originating Virtual Address when it is required to be absent
~~4~~*7*. TRANSMIT Disconnect-Request,
'Message ID' = (M2: any valid value),
'Originating Virtual Address' = D3, -- error condition, required to be absent
-- 'Destination Virtual Address' absent
'Destination Options' = (absent or a valid list of options),
-- 'Data Options' absent
~~5~~*8*. IF (receive a message) THEN
~~6~~*9*. RECEIVE BVLC-Result,
'Message ID' = M2,
-- 'Originating Virtual Address' absent
-- A valid message from a hub requires DVA to be absent
-- 'Destination Virtual Address' absent
'Destination Options' = (absent or a valid list of options),
-- 'Data Options' absent
'Result for BVLC Function' = X'08', -- Disconnect-Request
'Result Code' = X'01', -- NAK
'Error Header Marker' = X'00', -- not a header option problem
'Error Class' = (COMMUNICATION),
'Error Code' = (HEADER_ENCODING_ERROR,
INCONSISTENT_PARAMETERS,
PARAMETER_OUT_OF_RANGE or OTHER)
~~7~~*10*. CHECK(that the IUT did not process the request)

-- Inclusion of a 'Destination Virtual Address when it is required to be absent
~~8~~*11*.TRANSMIT Disconnect-Request,
'Message ID' = (M3: any valid value),
-- 'Originating Virtual Address' absent,
'Destination Virtual Address' = (IUT's VMAC), -- error condition, required to be absent
'Destination Options' = (absent or a valid list of options),
-- 'Data Options' absent
~~9~~*12*.IF (receive a message) THEN
~~10~~*13* RECEIVE BVLC-Result,
'Message ID' = M3,
-- 'Originating Virtual Address' absent
-- 'Destination Virtual Address' absent
'Destination Options' = (absent or a valid list of options),
-- 'Data Options' absent
'Result for BVLC Function' = X'08', -- Disconnect-Request

© 2026 by BACnet International. All rights reserved.

'Result Code' = X'01', -- NAK
'Error Header Marker' = X'00', -- not a header option problem
'Error Class' = (COMMUNICATION),
'Error Code' = (HEADER_ENCODING_ERROR,
INCONSISTENT_PARAMETERS,
PARAMETER_OUT_OF_RANGE or OTHER)

~~11~~*14*. CHECK(that the IUT did not process the request)

-- A truncated message

~~12~~*15*. TRANSMIT Encapsulated-NPDU,
'Message ID' = (M4: any valid value),
--'Originating Virtual Address' absent
'Destination Virtual Address' = (IUT's VMAC),
-- 'Destination Options' absent
-- 'Data Options' absent
-- no NPDU included in the message

~~13~~*16*. RECEIVE BVLC-Result,
'Message ID' = M4,
'Originating Virtual Address' = (IUT's VMAC),
-- 'Destination Virtual Address' absent
'Destination Options' = (absent or a valid list of options),
-- 'Data Options' absent
'Result for BVLC Function' = X'01', -- Encapsulated-NPDU
'Result Code' = X'01', -- NAK
'Error Header Marker' = X'00', -- not a header option problem
'Error Class' = COMMUNICATION,
'Error Code' = MESSAGE_INCOMPLETE | PAYLOAD_EXPECTED

~~14~~*17*. CHECK(that the IUT did not process the request)

-- A message with extra octets added on

~~15~~*18*. TRANSMIT Disconnect-Request,
'Message ID' = (M5: any valid value),
-- 'Originating Virtual Address' absent
-- 'Destination Virtual Address' absent
-- 'Destination Options' absent
-- 'Data Options' absent
(extra octets) = ({ X'C1', --a bunch of octets that look like valid data options.
X'BF0003000012',
X'3F0003000034'})

~~16~~*19*. RECEIVE BVLC-Result,
'Message ID' = M5,
-- 'Originating Virtual Address' absent
-- 'Destination Virtual Address' absent
'Destination Options' = (absent or a valid list of options),
-- 'Data Options' absent
'Result for BVLC Function' = X'08', -- Disconnect-Request
'Result Code' = X'01', -- NAK
'Error Header Marker' = X'00', -- not a header option problem
'Error Class' = COMMUNICATION,
'Error Code' = UNEXPECTED_DATA

~~17~~*20*. CHECK(that the IUT did not process the request)

-- A truncated broadcast message

~~18~~*21*. TRANSMIT Encapsulated-NPDU,
DESTINATION = GLOBAL_BROADCAST,
'Message ID' = (M6: any valid value),
-- 'Originating Virtual Address' absent,
'Destination Virtual Address' = (local broadcast VMAC),
-- 'Destination Options' absent
'Data Options' = ({ X'E1' -- Secure Path,

© 2026 by BACnet International. All rights reserved.

more follows (but none are present)
})

~~19~~*22*. CHECK(that the IUT does not send a BVLC-Result, did not process the message, nor route the message)

© 2026 by BACnet International. All rights reserved.

BTL-26.1 cr-imp1-6: Update COV Subscription Support Testing [BTLWG-1801, CR-0604]

Overview:

Test 9.10.1.11 assumes a device supports 5 COV subscriptions in a single object. A device must support 5 COV subscriptions in any combination of objects.

Changes:

Checklist Changes

None

Test Plan Changes

[Change all references of 9.10.1.11 from 135.1-2025 to BTL]

Specified Test Changes

9.10.1.11 Ensuring 5 Concurrent COV Subscribers

Reason For Change: Clarified the Test Concept and added Test Conditionality. Changed test to support 5 subscriptions from any number of objects.

Purpose: This test case verifies that the IUT can support 5 concurrent subscriptions.

Test Concept: Have the TD subscribe with 5 different process identifiers, *V1 through V5 to monitored objects O1 through O5.*~~V_1 through V_5 for monitored object O1.~~ *After each subscription, verify the IUT initiates the* ~~a~~ *corresponding notification* ~~is sent~~. *Change the value of the monitored object and verify the IUT initiates the corresponding notification.*~~that all 5 notifications were sent.~~ ~~, and then check to ensure that 5 notifications are sent when the monitored object changes.~~

Test Conditionality: The IUT should not have any subscriptions at the start of this test.

Notes to Tester: ~~The notification in step 3 can be received in any order by the TD.~~ *Object O1 through O5 can be the same object if 5 COV subscriptions are supported in a single object or different objects if no object supports 5 COV subscriptions.*

Test Steps:

1. REPEAT (*x = 1 to 5*~~X=V_1 to V_5~~) DO {
2. TRANSMIT SubscribeCOV-Request,
   'Subscriber Process Identifier' = *Vx*~~X~~,
   'Monitored Object Identifier' = *Ox*~~O1~~(~~any object supporting COV notifications~~),
   'Issue Confirmed Notifications' = TRUE | FALSE,
   'Lifetime' = (any valid value that will allow the subscription to outlast the test)
3. RECEIVE BACnet-SimpleACK-PDU
4. IF (~~if~~ confirmed notifications ~~were~~ requested) THEN
5. BEFORE **Notification Fail Time**
6. RECEIVE ConfirmedCOVNotification-Request,
   'Subscriber Process Identifier' = *Vx*~~X~~,
   'Initiating Device Identifier' = IUT,
   'Monitored Object Identifier' = *Ox*~~O1~~(~~the same object used in the subscription~~),
   'Time Remaining' = (any valid value),
   'List of Values' = (the initial Present_Value and initial Status_Flags)
7. TRANSMIT BACnet-SimpleACK-PDU

ELSE

© 2026 by BACnet International. All rights reserved.

8. BEFORE **Notification Fail Time**
9. RECEIVE UnconfirmedCOVNotification-Request,
'Subscriber Process Identifier' = *Vx~~X~~*,
'Initiating Device Identifier' = IUT,
'Monitored Object Identifier' = *Ox~~O1~~*~~(the same object used in the subscription)~~,
'Time Remaining' = (any valid value),
'List of Values' = (the initial Present_Value and initial Status_Flags)
}

~~2. MAKE (Present_Value = any value that differs from "initial Present_Value" such that a COV notification would be generated)~~

10. REPEAT (*x = 1 to 5*~~X=V1 to V5~~) DO {
11. MAKE (Ox, Present_Value change such that the IUT initiates the required COV notifications)
12. IF (~~if~~ confirmed notifications ~~were~~ requested) THEN
13. RECEIVE ConfirmedCOVNotification-Request,
'Subscriber Process Identifier' = *Vx~~X~~*,
'Initiating Device Identifier' = IUT,
'Monitored Object Identifier' = *Ox~~O1~~*~~(the same object used in the subscription)~~,
'Time Remaining' = (any valid value),
'List of Values' = (the new Present_Value and Status_Flags)
14. TRANSMIT BACnet-SimpleACK-PDU
ELSE
15. RECEIVE UnconfirmedCOVNotification-Request,
'Subscriber Process Identifier' = *Vx~~X~~*,
'Initiating Device Identifier' = IUT,
'Monitored Object Identifier' = *Ox~~O1~~*~~(the same object used in the subscription)~~,
'Time Remaining' = (any valid value),
'List of Values' = (the new Present_Value and Status_Flags)
}

© 2026 by BACnet International. All rights reserved.

BTL-26.1 cr-imp1-7: Initiating You-Are Messages and User Actions [BTLWG-1804, CR-0603]

Overview:

Test 9.43.1 currently assumes that the IUT automatically initiates a “YOU ARE” telegram upon receiving a “WHO-AM-I” request. However, CR-0603 clarifies that this behavior may depend on user interaction. For example, the tester (or user) may need to manually assign the Device Object Instance Number via the IUT’s software, as indicated in the Notes to Tester.

Changes:

Checklist Changes

None

Test Plan Changes

[Change all references of tests 135.1-2025 - 9.43.1 - Uses Who-Is to Configure Devices Supporting the Who-Am-I Service, 135.1-2025 - 8.45.3 - Can Unconfigure Devices, 135.1-2025 - 9.43.2 - Configures Other Device's Object Instance Number, 135.1-2025 - 8.45.2 - Configures Other Device's Object Instance Number and MAC Address to reference BTL]
]

Specified Test Changes

[Move test from 135.1 to BTL Specified Tests and modify as noted below]

9.43.1 Uses Who-Is to Configure Devices Supporting the Who-Am-I Service
Reason for Change: In accordance with CR-0603, the test procedure has been updated to include the time necessary for the user to provide the required input and to restrict the DESTINATION in compliance with Section 16.11.4.

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

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

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

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

Test Steps:
1. MAKE (the IUT transmit a Who-Is request to discover devices needing configuration)
2. RECEIVE
   DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   Who-Is Request,
   'Device Instance Range Low Limit' = 4194303,
   'Device Instance Range High Limit' = 4194303
3. TRANSMIT
   DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   Who-Am-I Request,
   'Vendor Identifier' = (TD's Vendor_Identifier),
   'Model Name' = (any valid Model Name),
   'Serial Number' = (any valid Serial Number)

~~4. BEFORE Internal Processing Fail Time~~

© 2026 by BACnet International. All rights reserved.

4. MAKE (the IUT to initiate a You-Are request to the TD)
5. RECEIVE
DESTINATION = TD ~~| GLOBAL BROADCAST | LOCAL BROADCAST | REMOTE BROADCAST~~,
You-Are Request,
'Vendor Identifier' = (TD's Vendor_Identifier),
'Model Name' = (the Model Name sent by TD in the previous step),
'Serial Number' = (the Serial Number sent by TD in the previous step),
'Device Identifier' = (Device, X),
'Device MAC Address' = (TD's MAC address, or absent)
6. IF (TD is not an MS/TP subordinate node)
7. TRANSMIT
DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
I-Am Request,
'Device Identifier' = (Device, X),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (TD's Vendor_Identifier)

[Move this test from 135.1 into BTL Specified Tests]

9.43.2 Configures Other Device's Object Instance Number
Reason for Change: In accordance with CR-0603, the test procedure has been updated to include the time necessary for the user to provide the required input and to restrict the DESTINATION in compliance with Section 16.11.4.

Purpose: To verify the IUT can configure another Device's object instance number using the You-Are service.

Test Concept: TD sends a Who-Am-I, and the IUT configures it with an appropriate Device object instance number (X).

Configuration Requirements: TD is configured with a MAC address but no Device object instance number. The IUT is not actively discovering devices.

Notes to Tester: The IUT may require the tester to specify a Device object instance number to assign TD, using the IUT's software. If the IUT requires this step to occur between it receiving the Who-Am-I request and sending a You-Are request, then the amount of time it takes the tester to enter this should not be counted towards the fail timer. The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. TRANSMIT
DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
Who-Am-I Request,
'Vendor Identifier' = (TD's Vendor_Identifier),
'Model Name' = (any valid Model Name),
'Serial Number' = (any valid Serial Number)
~~2. BEFORE Internal Processing Fail Time~~
2. MAKE (the IUT to initiate a You-Are request to TD)
3. RECEIVE
DESTINATION = TD ~~| LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST~~,
You-Are Request,
'Vendor Identifier' = (TD's Vendor_Identifier),
'Model Name' = (the Model Name sent by TD in the previous step),
'Serial Number' = (the Serial Number sent by TD in the previous step),
'Device Identifier' = (Device, X),
'Device MAC Address' = (TD's MAC address, or absent)
4. IF (TD is not an MS/TP subordinate node)
5. TRANSMIT
DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
I-Am Request,
'Device Identifier' = (Device, X)
'Max APDU Length Accepted' = (any valid value),

© 2026 by BACnet International. All rights reserved.

'Segmentation Supported' = (any valid value)

[In BTL Specified Tests, modify this test]

8.45.1 Configures Other Device's MAC Address

Reason for Change: wording.

Purpose: To verify the IUT can configure another device's MAC address using the You-Are service without relying on the Who-Am-I service.

Test Concept: The IUT configures TD with an appropriate MAC address without TD first sending a Who-Am-I.

Configuration Requirements: TD's Device object is configured with a known Device object instance number and no configured MAC address.

Notes to Tester: The IUT may require the tester to specify all the parameters needed to configure TD with You-Are, using the IUT's software. The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. MAKE (the IUT *initiate a You-Are request* to the TD)
2. RECEIVE
   DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   You-Are Request,
   'Vendor Identifier' = (TD's Vendor_Identifier),
   'Model Name' = (TD's Model_Name),
   'Serial Number' = (TD's Serial_Number),
   ~~'Device Identifier' = (TD's Device object),~~
   'Device MAC Address' = (an appropriate MAC address)
3. IF (TD is not an MS/TP subordinate node)
4. TRANSMIT
   DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
   I-Am Request,
   'Device Identifier' = (TD's Device object)
   'Max APDU Length Accepted' = (any valid value),
   'Segmentation Supported' = (any valid value),
   'Vendor Identifier' = (TD's Vendor_Identifier)

[Move this test from 135.1 into BTL Specified Tests]

8.45.2 Configures Other Device's Object Instance Number and MAC Address

Reason for Change: Fail time is not necessary for this user action

Purpose: To verify the IUT can configure another Device's object instance number and MAC address using the You-Are service without relying on the Who-Am-I service.

Test Concept: The IUT configures TD with an appropriate Device object instance number (X) and MAC address without TD first sending a Who-Am-I.

Configuration Requirements: TD is not configured with a Device object instance number and has no configured MAC address.

Notes to Tester: The IUT will require the tester to specify all the parameters needed to configure TD with You-Are, using the IUT's software. The destination address used by TD shall be selected such that the IUT will receive the messages.

Test Steps:

1. MAKE (the IUT *initiate a You-Are request to the* TD)
~~2. BEFORE Internal Processing Fail Time~~
2~~3~~. RECEIVE

© 2026 by BACnet International. All rights reserved.

DESTINATION = LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
You-Are Request,
'Vendor Identifier' = (TD's Vendor_Identifier),
'Model Name' = (TD's Model_Name),
'Serial Number' = (TD's Serial_Number),
'Device Identifier' = (Device, X),
'Device MAC Address' = (any valid MAC address)

34. IF (TD is not an MS/TP subordinate node)
45. TRANSMIT
DESTINATION = IUT | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
I-Am Request,
'Device Identifier' = (Device, X),
'Max APDU Length Accepted' = (any valid value),
'Segmentation Supported' = (any valid value),
'Vendor Identifier' = (TD's Vendor_Identifier)'Vendor Identifier' = (TD's Vendor_Identifier)

[Move this test from 135.1 into BTL Specified Tests]

8.45.3 Can Unconfigure Devices
Reason for Change: Fail time is not necessary for this user action

Purpose: To verify the IUT can unconfigure a device using the You-Are service.

Test Concept: The IUT transmits a You-Are with 'Device Identifier' = (Device, 4194303) in order to unconfigure TD.

Test Steps:

1. MAKE (the IUT *to initiate a You-Are request to* unconfigure the TD)
~~2. BEFORE Internal Processing Fail Time~~
23. RECEIVE
DESTINATION = TD | LOCAL BROADCAST | GLOBAL BROADCAST | REMOTE BROADCAST,
You-Are Request,
'Vendor Identifier' = (TD's Vendor_Identifier),
'Model Name' = (TD's Model_Name),
'Serial Number' = (TD's Serial_Number),
'Device Identifier' = (Device, 4194303),
'Device MAC Address' = (any valid MAC address, or absent)

© 2026 by BACnet International. All rights reserved.

BTL-26.1 cr-imp1-8: Improve Testing of Writing Stream Based File [BTLWG-1794, CR-0601]

Overview:

135.1-2025 - 9.12.1.2.1 - Reading an Entire Stream-Based File
Test conditionality updated for the scenario.

135.1-2025 - 9.13.1.2.1-Writing an Entire Stream Based File
Correction in steps 4 and 6 of '9.13.1.2.1-Writing an Entire Stream Based File' in case file_size = 0

Changes:

Checklist Changes

None

Test Plan Changes

3.61 File Object

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.

135.1-2025 - 9.12.1.2.1 - Reading an Entire Stream-Based File		
	Test Conditionality	~~Must be executed.~~ *This test shall be skipped if File_Size can only equal 0*
	Test Directives	Apply to a file not used for Backup & Restore or BACnet/SC Certificate Exchange.
	Testing Hints	
~~**135.1-2025**~~***BTL*** **9.13.1.2.1-Writing an Entire Stream Based File:** No Change in Test Plan		

Specified Test Changes

9.13.1.2.1-Writing an Entire Stream Based File:

Reason for Change: *Update in Step 4 and Step 6 considering possibility that File_size can be 0.*

Purpose: To verify that the IUT correctly responds to a request to write an entire file.
Test Concept: The tests consist of modifying the contents of the files using the AtomicWriteFile service and verifying that the appropriate changes to the file data took place.

Configuration Requirements: The manufacturer shall provide appropriate test data to write to these files or sufficient information to permit the tester to construct the test data. The file objects shall be configured with initial data that differs from the test data. In the test procedures, "X" will designate the File object identifier and Z the File Start Position' initialized at "1" at the beginning. When performing the AtomicWriteFile services, a Maximum Write Data Length (MWDL) shall be calculated before starting the test. These values shall be used during the test. MWDL shall be 21 less than the minimum of the TD's maximum transmittable APDU length and the IUT's Max_APDU_Length_Accepted.

Test Steps:

© 2026 by BACnet International. All rights reserved.

1. VERIFY Read_Only = FALSE
2. WRITE Archive = TRUE
3. VERIFY File_Access_Method = STREAM ACCESS
~~4. IF (File_Size is not equal to the size of the test file) THEN~~
~~WRITE File_Size = 0~~
4. IF (File_Size is not equal to the size of the test file AND File_Size != 0) THEN
WRITE File_Size = 0
5. REPEAT Z = (0 through the file size, in increments of MWDL) DO {
TRANSMIT AtomicWriteFile-Request
'File Identifier' = X
'File Start Position' = Z
'File Data' = (file contents, the number of octets being the lesser of (file size - Z)
and MWDL)
RECEIVE AtomicWriteFile-ACK
'File Start Position' = Z
}
6. VERIFY File_Size = ~~(file size of the test data)~~ *(file size of the test data or a value of 0)*
7. VERIFY Modification_Date = (the current date and time)
8. VERIFY ARCHIVE = FALSE

© 2026 by BACnet International. All rights reserved.

BTL-26.1 cr-imp1-9: Wrong Message Id Requirement in Heartbeat Tests [BTLWG-1822, CR-0607]

Overview:

Wrong Message-ID in tests:
12.5.1.1.16
12.5.1.2.10
12.5.2.2.6

Changes were also made to simplify testing.

Changes:

Checklist Changes

None

Test Plan Changes

[In Clause 9.9.2, change 135.1-2025 - 12.5.1.2.10 to BTL - 12.5.1.2.10]
[In Clause 9.9.3, change 135.1-2025 - 12.5.2.2.6 to BTL - 12.5.2.2.6]

Specified Test Changes

[Update BTL Specified Test 12.5.1.1.16]

12.5.1.1.16 Heartbeat-Request Initiation Test

Reason for Change: modified per Addendum 135-2020cc-1. Changed test to just check the heartbeat interval.

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

Test Concept: With the IUT connected to the BACnet/SC network, *make the IUT stop initiating requests except heartbeat requests,* wait *up to* the IUT's configured heart-beat interval plus 10 seconds, and verify ~~that~~ the IUT sent a Heartbeat-Request. *Wait the IUT's configured heart-beat interval plus 10 seconds and verify the IUT sent a Heartbeat-Request at the correct interval.* ~~, 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. MAKE (The IUT stop initiating any requests to the TD except for Heartbeat-Requests)
2. BEFORE IUT's heartbeat interval + 10s
3. RECEIVE Heartbeat-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
4. TRANSMIT Heartbeat-ACK,
'Message ID' = M1,
-- 'Originating Virtual Address' absent
-- 'Destination Virtual Address' absent

© 2026 by BACnet International. All rights reserved.

'Destination Options' = (absent or any valid value),
-- 'Data Options' absent
5. T1 = (the time the Heartbeat-Request was received)
6. BEFORE IUT's heartbeat interval + 10s
7. RECEIVE Heartbeat-Request,
'Message ID' = (M2: any valid value including M1),
-- 'Originating Virtual Address' absent
-- 'Destination Virtual Address' absent
'Destination Options' = (absent or any valid value),
-- 'Data Options' absent
8. TRANSMIT Heartbeat-ACK,
'Message ID' = M2,
-- 'Originating Virtual Address' absent
-- 'Destination Virtual Address' absent
'Destination Options' = (absent or any valid value),
-- 'Data Options' absent
9. T2 = (the time the Heartbeat-Request was received)
10. CHECK (T2 - T1 ~= heartbeat interval)
~~1. REPEAT N = (1..Z) DO {~~
~~2. TRANSMIT Encapsulated-NPDU,~~
~~'Message ID' = (M1: 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~~
~~3. RECEIVE Encapsulated-NPDU,~~
~~'Message ID' = M1,~~
~~'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)~~
~~}~~
~~4. BEFORE ½ of IUT's heartbeat interval + 10s~~
~~5. RECEIVE Heartbeat-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~~
~~6. TRANSMIT Heartbeat-ACK,~~
~~'Message ID' = M2,~~
~~'Originating Virtual Address' absent~~
~~'Destination Virtual Address' absent~~
~~'Destination Options' = (absent or any valid value),~~
~~'Data Options' absent~~

[Modify 135-2025 - 12.5.1.2.10]

12.5.1.2.10 Node Heartbeat-Request Initialization Failure Test

© 2026 by BACnet International. All rights reserved.

Reason for Change: Changed the test to just test that the IUT initiates a Disconnect request when the heartbeat request is not acknowledged.

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. *Make the IUT initiate a Heartbeat-Request and make sure the TD does not respond to the request. Wait the vendor specified timeout for the IUT to initiate a Disconnect-Request.* ~~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.

Notes to Tester: To make the IUT initiate a Heartbeat-Request, make the TD not initiate or respond to any requests from the IUT.

Test Steps:

~~1. REPEAT N = (1..Z) {~~
~~2. 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~~
~~3. 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)~~
~~4. 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~~
~~5. BEFORE HB / 2 + 10s~~

1. BEFORE heartbeat timeout + 10s
2. 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)
4. 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

© 2026 by BACnet International. All rights reserved.

[Modify 135-2025 - 12.5.2.2.6]

12.5.2.2.6 Hub Heartbeat-Request Initialization Failure Test

Reason for Change: Changed the test to just test that the IUT initiates a Disconnect request when the heartbeat request is not acknowledged.

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. *Make the IUT initiate a Heartbeat-Request and make sure the TD does not respond to the request. Wait the vendor specified timeout for the IUT to initiate a Disconnect-Request.* ~~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.~~*If the IUT's hub function does not have a mechanism to determine if the connection or the connection peer might not be alive, this test shall be skipped.*

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) {~~
~~2. 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~~
~~3. 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)~~
~~4. 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~~
~~5. BEFORE HB / 2 + 10s~~
1. MAKE (the IUT initiate a Heartbeat-Request)
2. 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)
4. RECEIVE Disconnect-Request,
'Message ID' = (~~M:~~ any valid value)

© 2026 by BACnet International. All rights reserved.

-- 'Originating Virtual Address' absent
-- 'Destination Virtual Address' absent
'Destination Options' = (absent or any valid value),
-- 'Data Options' absent

© 2026 by BACnet International. All rights reserved.

BTL-26.1 cr-imp1-10: Update Test Directives for Router Tests [BTLWG-932, CR-0215]

Overview:

Only the test directives for the tests 10.2.4.6 and 10.2.4.7 are to be modified as per CR-0215 and there is no change in the tests.

In response to CR-0215, the BTL agreed to improve the Test Directives for these two tests to be more clear in how the network setup relates to the network setup from 135.1.

Changes:

Checklist Changes

None

Test Plan Changes

10.1.2Routes Packets between Physical BACnet LANs

135.1-2025 - 10.2.4.6 - Remote Broadcast from a Remote Device to a Directly-Connected Network		
	Test Conditionality	Must be executed.
	Test Directives	~~Repeat twice for each LAN type supported by the device, once with the source router connected to the LAN and once with the LAN as the destination network for the remote broadcast.~~ *Must be executed a sufficient number of times to ensure that each supported data link is tested at least once in each of the 'Network 1' and 'Network 2' positions in 135.1, Figure 10-1.* Also, vary the length of SADR (1-6 bytes).
	Testing Hints	

135.1-2025 - 10.2.4.7 - Remote Broadcast from a Remote Device to a Remote Network		
	Test Conditionality	Must be executed.
	Test Directives	~~Repeat twice for each LAN type supported by the device, once with the source router connected to the LAN and once with the LAN as the destination network for the remote broadcast.~~ *Must be executed a sufficient number of times to ensure that each supported data link is tested at least once in each of the 'Network 1' and 'Network 2' positions in 135.1, Figure 10-1.* Also, vary the length of SADR (1-6 bytes).
	Testing Hints	

Specified Test Changes

None.

© 2026 by BACnet International. All rights reserved.