



BACnet[®] TESTING LABORATORIES ADDENDA

Addendum log to BTL Test Package 18.1

**Revision 4
Revised 8/17/2021**

Approved by the BTL Working Group on April 1, 2021.
Approved by the BTL Working Group Voting Members on August 2, 2021.
Published on August 19, 2021.

[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-18.1-log-1: Negative ReadRange Test Specifying 0 for ByPosition [BTLWG-501]..... 2

BTL-18.1-log-2: Add Delay to Tests 7.3.2.25.2 and 7.3.2.25.4 [BTLWG-873]..... 4

BTL-18.1-log-3: Add Ability to claim Intrinsic Reporting in Event Log Object [BTLWG-890] 6

BTL-18.1-log-4: ReadRange Test Improvements [BTLWG-980] 8

In the following document, language to be added to existing clauses within the BTL Test Package 18.1 is indicated through the use of *italics*, while deletions are indicated by ~~striketrough~~. Where entirely new subclauses are proposed to be added, plain type is used throughout

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

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

BTL-18.1-log-1: Negative ReadRange Test Specifying 0 for ByPosition [BTLWG-501]**Overview:**

Add a negative ReadRange test specifying 0 then specifying beyond the end, in ByPosition requests. <http://www.bacnet.org/Addenda/Add-135-2012bg.pdf> in bg-7 at Protocol_Revision 18 has clarified this in clause 15.8.1.1.4.1.1: "If the item with the index specified in this parameter does not exist, then no items match the criteria for being read and returned, regardless of the value of the 'Count' parameter."

15.8.1.1.4.1.1 Reference Index

The 'Reference Index' parameter specifies the index of the first (if 'Count' is positive) or last (if 'Count' is negative) item to be read. If the item with the index specified in this parameter does not exist, then no items match the criteria for being read and returned, regardless of the value of the 'Count' parameter.

Changes:

[In BTL Test Plan, update section 5.14.2 Supports all forms of ReadRange to include new test]

BTL - 9.21.2.X6 - Reading a Range of Items that do not Exist (by Position)		
	Test Conditionality	If IUT claims Protocol_Revision less than 18, then this test shall be skipped.
	Test Directives	Repeat the test for 'Reference Index' = 0, 'Reference Index' > Buffer_Size and 'Reference Index' between Record_Count and Buffer_Size.
	Testing Hints	

[In BTL Test Plan, update section 5.15.2 Supports all forms of ReadRange to include new test]

BTL - 9.21.2.X6 - Reading a Range of Items that do not Exist (by Position)		
	Test Conditionality	If IUT claims Protocol_Revision less than 18, then this test shall be skipped.
	Test Directives	Repeat the test for 'Reference Index' = 0, 'Reference Index' > Buffer_Size and 'Reference Index' between Record_Count and Buffer_Size.
	Testing Hints	

[In BTL Test Plan, update section 7.3.2 Supports all forms of ReadRange to include new test]

BTL - 9.21.2.X6 - Reading a Range of Items that do not Exist (by Position)		
	Test Conditionality	If IUT claims Protocol_Revision less than 18, then this test shall be skipped.
	Test Directives	Repeat the test for 'Reference Index' = 0, 'Reference Index' > Buffer_Size and 'Reference Index' between Record_Count and Buffer_Size.
	Testing Hints	

[In BTL Test Plan, update section 7.7.2 Supports all forms of ReadRange to include new test]

BTL - 9.21.2.X6 - Reading a Range of Items that do not Exist (by Position)		
	Test Conditionality	If IUT claims Protocol_Revision less than 18, then this test shall be skipped.
	Test Directives	Repeat the test for 'Reference Index' = 0, 'Reference Index' > Buffer_Size and 'Reference Index' between Record_Count and Buffer_Size.
	Testing Hints	

[In BTL Specified Tests, add new Test]

9.21.2.X6 Reading a Range of Items that do not Exist (Using by Position)

Purpose: To verify that the IUT correctly responds to a ReadRange service request when there are no items within the specified criteria.

Test Concept: A ReadRange request is transmitted by the TD requesting a range of items known not to be in the Log_Buffer. The IUT shall respond by returning an empty list.

Test Steps:

1. TRANSMIT ReadRange-Request,

'Object Identifier' =	(the log object configured for this test),
'Property Identifier' =	Log_Buffer,
'Reference Index' =	(any value that will result in no items being present)
'Count' =	(any non-zero number)
2. RECEIVE ReadRange-ACK,

'Object Identifier' =	(the log object configured for this test),
'Property Identifier' =	Log_Buffer,
'Result flags' =	{FALSE, FALSE, FALSE},
'Item Count' =	0,
'Item Data' =	(an empty list)
'First Sequence Number' =	(should be absent)

Test Example 1 (using index that does not exist):

1. TRANSMIT ReadRange-Request,

'Object Identifier' =	(the log object configured for this test),
'Property Identifier' =	Log_Buffer,
'Reference Index' =	0
'Count' =	5
2. RECEIVE ReadRange-ACK,

'Object Identifier' =	(the log object configured for this test),
'Property Identifier' =	Log_Buffer,
'Result flags' =	{FALSE, FALSE, FALSE},
'Item Count' =	0,
'Item Data' =	(an empty list)

Test Example 2 (using index that does not exist):

1. TRANSMIT ReadRange-Request,

'Object Identifier' =	(the log object configured for this test),
'Property Identifier' =	Log_Buffer,
'Reference Index' =	Index= Buffer_Size + 1
'Count' =	-20
2. RECEIVE ReadRange-ACK,

'Object Identifier' =	(the log object configured for this test),
'Property Identifier' =	Log_Buffer,
'Result flags' =	{FALSE, FALSE, FALSE},
'Item Count' =	0,
'Item Data' =	(an empty list)

BTL-18.1-log-2: Add Delay to Tests 7.3.2.25.2 and 7.3.2.25.4 [BTLWG-873]

Overview:

CR-0268 response specified that in each of tests 7.3.2.26.X2 and 7.3.2.26.X4 a delay is added after the first step. The numbering of both tests has been changed to 7.3.2.25.2 Remote Logging of Notifications and 7.3.2.25.4 Remote Logging of ACK_NOTIFICATIONs

Changes:

[In BTL Specified Tests, modify tests 7.3.2.25.2 and 7.3.2.25.4]

Reason for Change: Add delay after Step 1.

Purpose: To verify that the IUT correctly collects and represents the Notifications which it receives.

Test Concept: Make TD send multiple event notification messages. Use ReadRange to retrieve the events from an Event Log or perhaps from multiple Event Logs in the IUT, and compare the two representations.

Configuration Requirements: LO1 is an Event Log object in IUT which logs the event types which are sent. Stop_When_Full in LO1 shall be FALSE or absent.

Test Steps:

1. WRITE Enable = TRUE
2. *WAIT Internal Processing Fail Time*
3. TRANSMIT ConfirmedEventNotification-Request,
 - 'Process Identifier' = (any valid process identifier),
 - 'Initiating Device Identifier' = TD,
 - 'Event Object Identifier' = (any valid object identifier),
 - 'Time Stamp' = (T1, any valid timestamp),
 - 'Notification Class' = (any valid notification class),
 - 'Priority' = (any valid priority),
 - 'Event Type' = (any standard event type),
 - 'Message Text' = (optional, any valid message text),
 - 'Notify Type' = ALARM | EVENT,
 - 'AckRequired' = TRUE | FALSE,
 - 'From State' = (state S1, any valid state for this event type),
 - 'To State' = (state S2, any valid state for this event type that can follow S1),
 - 'Event Values' = (any values appropriate to the event type)
4. RECEIVE BACnet-SimpleACK-PDU
5. TRANSMIT ConfirmedEventNotification-Request,
 - 'Process Identifier' = (any valid process identifier),
 - 'Initiating Device Identifier' = IUT,
 - 'Event Object Identifier' = (any valid object identifier),
 - 'Time Stamp' = (T2, any valid timestamp),
 - 'Notification Class' = (any valid notification class),
 - 'Priority' = (any valid priority),
 - 'Event Type' = (any standard event type),
 - 'Message Text' = (optional, any valid message text),
 - 'Notify Type' = ALARM | EVENT,
 - 'AckRequired' = TRUE | FALSE,
 - 'From State' = (state S3, any valid state for this event type),
 - 'To State' = (state S4, any valid state for this event type that can follow S3),
 - 'Event Values' = (any values appropriate to the event type)
6. RECEIVE BACnet-SimpleACK-PDU
7. READ RC = LO1, Record_Count
8. TRANSMIT ReadRange-Request,
 - 'Object Identifier' = LO1,
 - 'Property Identifier' = Log_Buffer,
 - 'Reference Index' = RC,
 - 'Count' = -2

9. RECEIVE ReadRange-ACK,
 'Object Identifier' = LO1,
 'Property Identifier' = Log_Buffer,
 'Result Flags' = {FALSE, ?, FALSE},
 'Item Count' = 2,
 'Item Data' = (logged data that matches the information received in steps 2 and 4,
 except that Process_Identifier can be any value and is not required to
 match)
10. CHECK (that the events were logged in the order in which they were received)

Notes to Tester: When the UnconfirmedEventNotification service is used instead of the ConfirmedEventNotification service, the test shall skip the steps in which a *BACnet-SimpleACK-PDU* is expected.

7.3.2.25.4 Remote Logging of ACK_NOTIFICATIONs

Reason for Change: Add delay after Step 1.

Purpose: To verify that the IUT correctly collects and represents ACK_NOTIFICATIONs which it receives.

Test Concept: Send an ACK_NOTIFICATION to the IUT. Use ReadRange to retrieve that same event from an Event Log, and compare the two representations.

Configuration Requirements: LO1 is an Event Log object in IUT which logs ACK_NOTIFICATIONs. Stop_When_Full in LO1 shall be FALSE or absent.

Test Steps:

1. WRITE Enable = TRUE
2. WAIT Internal Processing Fail Time
3. TRANSMIT ConfirmedEventNotification-Request,
 'Process Identifier' = (any valid process identifier),
 'Initiating Device Identifier' = IUTID,
 'Event Object Identifier' = (any valid object identifier),
 'Time Stamp' = (T1, any valid timestamp),
 'Notification Class' = (any valid notification class),
 'Priority' = (any valid priority),
 'Event Type' = (any standard event type),
 'Message Text' = (optional, any valid message text),
 'Notify Type' = ACK_NOTIFICATION,
 'From State' = (state S1, any valid state for this event type)
4. RECEIVE BACnet-SimpleACK-PDU
5. READ RC = LO1, Record_Count
6. TRANSMIT ReadRange-Request,
 'Object Identifier' = LO1,
 'Property Identifier' = Log_Buffer,
 'Reference Index' = RC,
 'Count' = -1
7. RECEIVE ReadRange-ACK,
 'Object Identifier' = LO1,
 'Property Identifier' = Log_Buffer,
 'Result Flags' = {FALSE, ?, FALSE},
 'Item Count' = 1,
 'Item Data' = (logged data that matches the information received in step 2,
 except that Process_Identifier can be any value and is not required to match)

Notes to Tester: When the UnconfirmedEventNotification service is used instead of the ConfirmedEventNotification service, the test shall skip the step in which a *BACnet-SimpleACK-PDU* is expected.

BTL-18.1-log-3: Add Ability to claim Intrinsic Reporting in Event Log Object [BTLWG-890]**Overview:**

There is no provision in the checklist to claim intrinsic or algorithmic reporting of Event Log Object.

Changes:

[In BTL Checklist, Add BUFFER_READY Notification entry]

Event Log Object		
	R	Base Requirements
	C ¹	Supports AE-EL-I-B
	C ¹	Supports AE-EL-E-B
	O	Supports BUFFER_READY Notification
¹ At least one of these options is required if the IUT supports the Event Log object type.		

[In BTL Test Plan, Add new entry to section 3.22 Event Log Object]

3.22 Event Log Object

...

3.22.4 Supports BUFFER_READY Notification

135.1-2019 - 8.4.7 - BUFFER_READY Tests (ConfirmedEventNotification)		
	Test Conditionality	If BUFFER_READY Notification is tested through 135.1-2019 - 8.5.7, then this test can be skipped.
	Test Directives	The 'Event Object Identifier' in this test must be either an Event Log object contained in the IUT or an Event Enrollment object in the IUT monitoring the Event Log Object contained in the IUT.
	Testing Hints	
135.1-2019 - 8.5.7 - BUFFER_READY Tests (UnconfirmedEventNotification)		
	Test Conditionality	If BUFFER_READY Notification is tested through 135.1-2019 - 8.4.7, then this test can be skipped.
	Test Directives	The 'Event Object Identifier' in this test must be either an Event Log object contained in the IUT or an Event Enrollment object in the IUT monitoring the Event Log Object contained in the IUT.
	Testing Hints	
135.1-2019 - 7.3.1.10.2 - Event Enable Tests for TO_NORMAL only Algorithms		
	Test Conditionality	If the IUT cannot be configured to meet the configuration requirements then this test shall be skipped.
	Test Directives	The 'Event Object Identifier' in this test must be either an Event Log object contained in the IUT or an Event Enrollment object in the IUT monitoring the Event Log Object contained in the IUT.
	Testing Hints	
135.1-2019 - 7.3.1.12 - Notify_Type Test		
	Test Conditionality	If the IUT cannot be configured to meet the 135.1-2019 configuration requirements then this test shall be skipped.
	Test Directives	The 'Event Object Identifier' in this test must be either an Event Log object contained in the IUT or an Event Enrollment object in the IUT monitoring the Event Log Object contained in the IUT.
	Testing Hints	
135.1-2019 - 7.3.2.24.10 - Notification_Threshold Test		
	Test Conditionality	If the IUT cannot be configured to meet the 135.1-2019 configuration requirements then this test shall be skipped.
	Test Directives	The 'Event Object Identifier' in this test must be either an Event Log object contained in the IUT or an Event Enrollment object in the IUT monitoring the Event Log Object contained in the IUT.
	Testing Hints	

135.1-2019 - 7.3.2.24.17 - Last Notify Record Test		
	Test Conditionality	If the IUT cannot be configured to meet the configuration requirements then this test shall be skipped.
	Test Directives	The 'Event Object Identifier' in this test must be either an Event Log object contained in the IUT or an Event Enrollment object in the IUT monitoring the Event Log Object contained in the IUT.
	Testing Hints	
135.1-2019 - 7.3.2.24.18 - Records Since Notification Test		
	Test Conditionality	If the IUT cannot be configured to meet the 135.1-2019 configuration requirements then this test shall be skipped.
	Test Directives	The 'Event Object Identifier' in this test must be either an Event Log object contained in the IUT or an Event Enrollment object in the IUT monitoring the Event Log Object contained in the IUT.
	Testing Hints	

BTL-18.1-log-4: ReadRange Test Improvements [BTLWG-980]**Overview:**

The BTL Specified Tests document contains additional ReadRange tests for clause 21 of 135.1 which can apply to any object type. But in clause 21, there is a base assumption that the tests are to be applied to a Trend Log object. The Test Plan applies the 9.21 tests to object types other than the Trend Log. Changes to the Test Plan and Specified Tests are needed to support the other log object types.

In making these changes, the testing for logging of status / failures in logging objects is moved out of the ReadRange testing and into 7.3.2.24 Logging Object Tests.

Changes:

Checklist Changes

None

Test Plan Changes

[Modify Test Plan references from 135.1-2019 to BTL for the following tests: 9.21.1.1, 9.21.1.2, 9.21.1.3, 9.21.1.6, 9.21.1.9, 9.21.1.10]

[Modify Test Plan References to 9.21.1.12 in 7.4.1 and 7.8.1]

135.1-2019 - 9.21.1.12 BTL - 7.3.2.24.X1 - Status/Failure Logging		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

[Replace all Test Plan entries referencing 9.21.1.11 as follows, except in 7.4.8]

[Replace <datatype> with the appropriate datatype for the section]

135.1-2019 - 9.21.1 - Positive ReadRange Service Execution Tests		
	Test Conditionality	Must be executed.
	Test Directives	Configure the log object referencing a property of <datatype>. Apply one of the following tests 9.21.1.1, 9.21.1.2, 9.21.1.3, 9.21.1.4, 9.21.1.4.1, 9.21.1.9 and 9.21.1.10
	Testing Hints	

[Modify Test Plan section 7.4.8.]

7.4.8 Is Able to Trend NULL Values

The IUT can be made to trend any type property that may change to a value of NULL.

135.1-2019 - 9.21.1 - Positive ReadRange Service Execution Tests:11 - Data Type Verification Test		
	Test Conditionality	Must be executed.
	Test Directives	Set Log DeviceObjectProperty to an external property that can be set to NULL value. Configure the log object referencing a property of <datatype>. Apply one of the following tests 9.21.1.1, 9.21.1.2, 9.21.1.3, 9.21.1.4, 9.21.1.4.1, 9.21.1.9 and 9.21.1.10
	Testing Hints	The best way to do this is to trend an entry in a priority array.

		Schedule_Default and Present_Value of the Schedule Object, Alarm_Values and Fault_Values of the CharacterString Value Object and Low_Diff_Limit in the Loop Object, are standard properties that should accept a written NULL.
--	--	--

[Add into Test Plan section 4.16.1 Execute Read Range Base Requirements]

BTL - 9.21.1.6 - Reading a Range of Items that do not Exist by Position		
	Test Conditionality	If the only list properties supported by the device is Log_Buffer, this test shall be skipped.
	Test Directives	Apply to a single non-empty list property that is not Log_Buffer.
	Testing Hints	

Test Changes

[Move clause 9.21 into BTL Specified Tests and modify]

9.21 ReadRange Execution Tests

This clause defines the tests necessary to demonstrate support for executing ReadRange service requests.

Dependencies: None.

BACnet Reference Clause: 15.8.

Configuration Requirements: *When testing a Log_Buffer property, the IUT shall be configured with a Trend Log logging object that contains a set of known log records.* ~~The IUT shall be configured with a Trend Log logging object that contains a set of known log records.~~ The TD must have exact knowledge of the ~~trend data~~ log records in order to evaluate the results of the tests. The value of the Enable property shall be FALSE so that the Log_Buffer does not change during the tests.

When testing a property other than the Log_Buffer, steps shall be taken to ensure that the value of the property does not change outside the control of the tester during the execution of the test.

The following sample log buffer is used as explanation for the tests in this section.

Sample Log_Buffer, (Trend Log, Instance 1)

Arbitrary Record Designation	Position (index)	Implied Sequence #	Timestamp (Date excluded for clarity)	LogDatum
a	1	16	13:01:00.00	log-status, buffer-purged
b	2	17	13:02:00.00	log-status, log-disabled = FALSE
c	3	18	13:05:00.00	real-value = 5.0
d	4	19	13:10:00.00	real-value = 10.0
e	5	20	13:15:00.00	real-value = 15.0
f	6	21	13:16:00.00	log-status, log-disabled = TRUE
g	7	22	13:21:00.00	log-status, log-disabled = FALSE
h	8	23	13:25:00.00	real-value = 25.0
i	9	24	13:30:00.00	real-value = 30.0
j	10	25	13:35:00.00	real-value = 35.0
k	11	26	13:36:00.00	log-status, log-disabled = TRUE

[Move clause 9.21.1.1 into BTL Specified Tests and modify]

9.21.1.1 Reading All Items in the List

Purpose: To verify that the IUT correctly responds to a ReadRange service request to return all of the available data items.

Test Concept: A list property, P is read using ReadRange by position with no range specified. It is verified that the complete list is returned.

Configuration Requirements: Property P is configured with a value that is small enough to be returned in a single ReadRange response. If the IUT cannot be configured in this manner, see the Notes to Tester.

Test Steps:

1. TRANSMIT ReadRange-Request,
 - 'Object Identifier' = (the ~~log~~ object configured for this test),
 - 'Property Identifier' = ~~Log_Buffer~~ P
2. RECEIVE Read-Range-ACK,
 - 'Object Identifier' = (the log object configured for this test),
 - 'Property Identifier' = Log_Buffer,
 - 'Result Flags' = {TRUE, TRUE, FALSE},
 - 'Item Count' = (the number of *entries in P* ~~trend records in the test object~~),
 - 'Item Data' = (all of the *entries in P* ~~trend records in the test object~~)

Notes to Tester: The ~~trend data~~ property P may have more items than can be returned in a single message. Under these circumstances 'Result Flags' will have the value {TRUE, FALSE, TRUE} and the 'Item Count' and 'Item Data' parameters would reflect the actual number of items that were able to be returned.

[Move clause 9.21.1.2 into BTL Specified Tests and modify]

9.21.1.2 Reading Items by Position with Positive Count

Purpose: To verify that the IUT correctly responds to a ReadRange service request to return items specified by indicating a position and the number of items after that position to return.

Test Concept: A ReadRange request is transmitted by the TD requesting a range of items known to be in the list property P the ~~Log_Buffer~~. This range is specified using the 'By Position' option and a positive value for 'Count'. The 'Reference Index' and 'Count' are selected so that the results can be conveyed in a single acknowledgment.

Configuration Requirements: A list property, P, is configured with N items.

Test Steps:

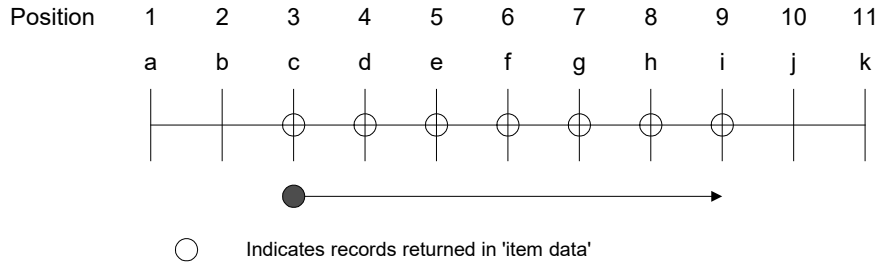
1. TRANSMIT ReadRange-Request,
 - 'Object Identifier' = (the ~~log~~ object configured for this test),
 - 'Property Identifier' = ~~Log_Buffer~~ P,
 - 'Reference Index' = (any value x: $1 \leq x \leq \text{Record_Count } N$),
 - 'Count' = (any value y \times : $0 < y \times \leq \text{Record_Count } N - x + 1$)
2. RECEIVE Read-Range-ACK,
 - 'Object Identifier' = (the ~~log~~ object configured for this test),
 - 'Property Identifier' = ~~Log_Buffer~~ P,
 - 'Result Flags' = {?, ?, FALSE},
 - 'Item Count' = y (the same value used in the 'Count' parameter in step 1),
 - 'Item Data' = (all of the *items* specified ~~trend records~~ in order of increasing position. The items specified include
the item at the index specified by x, plus (y-1) items following.)

Test Example (using the sample buffer at beginning of section):

1. TRANSMIT ReadRange-Request,
 - 'Object Identifier' = (Trend Log, Instance 1),
 - 'Property Identifier' = Log_Buffer,
 - 'Reference Index' = 3,
 - 'Count' = 7

2. RECEIVE ReadRange-ACK,

'Object Identifier'	= (Trend Log, Instance 1),
'Property Identifier'	= Log_Buffer,
'Result Flags'	= {FALSE, FALSE, FALSE},
'Item Count'	= 7,
'Item Data'	= Records < c, d, e, f, g, h, i > in that order.



[Move clause 9.21.1.3 into BTL Specified Tests and modify]

9.21.1.3 Reading Items by Position with Negative Count

Purpose: To verify that the IUT correctly responds to a ReadRange service request to return items specified by indicating a position and the number of items before that position to return.

Test Concept: A ReadRange request is transmitted by the TD requesting a range of items known to be in *the list property P* ~~the Log_Buffer~~. This range is specified using the 'By Position' option and a negative value for 'Count'. The 'Reference Index' and 'Count' are selected so that the results can be conveyed in a single acknowledgement.

Configuration Requirements: A list property, P, is configured with N items.

Test Steps:

1. TRANSMIT ReadRange-Request,

'Object Identifier' =	(the log object configured for this test),
'Property Identifier' =	Log_Buffer P,
'Reference Index' =	(any value x: $1 \leq x \leq N$ Record_Count),
'Count' =	(any value y: $y < 0$ AND $ y \leq x$)
2. RECEIVE ReadRange-ACK,

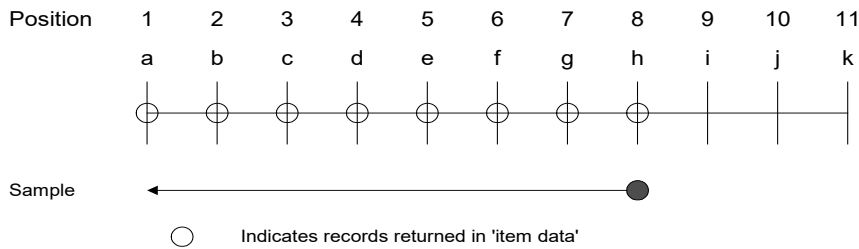
'Object Identifier' =	(the log object configured for this test),
'Property Identifier' =	Log_Buffer P,
'Result Flags' =	{?, ?, FALSE},
'Item Count' =	$ y $,
'Item Data' =	(all of the <i>items</i> specified trend-records in order of increasing position. The items specified include the item at the index specified by x, plus $ y -1$ items preceding.)

Test Example (using the sample buffer at beginning of section):

1. TRANSMIT ReadRange-Request,

'Object Identifier'	= (Trend Log, Instance 1),
'Property Identifier'	= Log_Buffer,
'Reference Index'	= 8,
'Count'	= -8
2. RECEIVE ReadRange-ACK,

'Object Identifier'	= (Trend Log, Instance 1),
'Property Identifier'	= Log_Buffer,
'Result Flags'	= {TRUE, FALSE, FALSE},
'Item Count'	= 8
'Item Data'	= Records < a, b, c, d, e, f, g, h > in that order.



[Move clause 9.21.1.5 into BTL Specified Tests and modify to show this test is being removed]

9.21.1.5 Reading Items by Time Range

Purpose: To verify that the IUT correctly responds to a ReadRange service request to return items specified by indicating a range of times that are to be included.

Test Concept: A ReadRange request is transmitted by the TD requesting a range of items known to be in the Log Buffer. This range is specified using the "Time Range" option. The 'Beginning Time' and 'Ending Time' are selected so that the results can be conveyed in a single acknowledgement.

Test Steps:

1. TRANSMIT ReadRange Request,
 - 'Object Identifier' = (the Trend Log log object configured for this test);
 - 'Property Identifier' = Log_Buffer;
 - 'Beginning Time' = (any value before the last time in the buffer);
 - 'Ending Time' = (any value > 'Beginning Time');
2. RECEIVE Read Range ACK,
 - 'Object Identifier' = (the Trend Log log object configured for this test);
 - 'Property Identifier' = Log_Buffer;
 - 'Result flags' = {TRUE, TRUE, FALSE};
 - 'Item Count' = (the number of trend records *items* meeting the specified criteria);
 - 'Item Data' = (all of the specified *items* trend records)

Notes to Tester: The first item returned shall be the first one in the buffer that has a timestamp newer (later time) than the time specified by the 'Beginning Time' parameter. The last item returned shall be the one with a timestamp older (earlier time) than or equal to the one specified by the 'Ending Time' parameter.

This clause removed.

[Move clause 9.21.1.6 into BTL Specified Tests and modify]

9.21.1.6 Reading a Range of Items that do not Exist by Position

Purpose: To verify that the IUT correctly responds to a ReadRange service request when there are no items within the specified by position range.

Test Concept: A ReadRange request is transmitted by the TD requesting a range of items known not to be in the list property *P* Log_Buffer. The IUT shall respond by returning an empty list.

Configuration Requirements: The list property, *P*, is configured with *N* items.

Test Steps:

1. TRANSMIT ReadRange-Request,
 - 'Object Identifier' = (the Trend Log log object configured for this test);
 - 'Property Identifier' = *P* Log_Buffer;
 - 'Reference Index' = (any value *x*: $x > N$);
 - 'Count' = (any value *y*: $y > 0$);
 - 'Beginning Time' = (any value that will result in a time interval for which there are no items present);
 - 'Ending Time' = (any value that will result in a time interval for which there are no items present)
2. RECEIVE Read-Range-ACK,

'Object Identifier' = (the ~~Trend-Log~~ *log* object configured for this test),
 'Property Identifier' = *P Log_Buffer*,
 'Result flags' = {TRUE, TRUE, FALSE},
 'Item Count' = 0,
 'Item Data' = (an empty list)

[Move clause 9.21.1.9 into BTL Specified Tests and modify]

9.21.1.9 Reading Items by Sequence with Positive Count

Purpose: To verify that the IUT correctly responds to a ReadRange service request to return items specified by indicating a sequence number and the number of items after that sequence to return.

Test Concept: A ReadRange request is transmitted by the TD requesting a range of items known to be in the Log_Buffer. This range is specified using the 'By Sequence' option and a positive value for 'Count'. The 'Reference Sequence Number' and 'Count' are selected so that the results can be conveyed in a single acknowledgment.

Test Steps:

1. TRANSMIT ReadRange-Request,
 - 'Object Identifier' = (the log object configured for this test),
 - 'Property Identifier' = Log_Buffer,
 - 'Reference Sequence Number' = (any value x : $(\text{Total_Record_Count} - \text{Record_Count} + 1) \leq x \leq (\text{Total_Record_Count} - y + 1)$),
 - 'Count' = (any value y : $0 < y \leq \text{Record_Count}$)
2. RECEIVE ReadRange-ACK,
 - 'Object Identifier' = (the log object configured for this test),
 - 'Property Identifier' = Log_Buffer,
 - 'Result Flags' = {?, ?, FALSE},
 - 'Item Count' = y ,
 - 'Item Data' = (All of the specified ~~trend~~-records in the order of increasing sequence number. The items specified are all items with the sequence number in the range of x through $(x+y-1)$ in that order).
 - 'First Sequence Number' = x

Test Example (using sample buffer at beginning of section):

1. TRANSMIT ReadRange-Request,
 - 'Object Identifier' = 20:1,
 - 'Property Identifier' = Log_Buffer,
 - 'Reference Sequence Number' = 16,
 - 'Count' = 11
2. RECEIVE ReadRange-ACK,
 - 'Object Identifier' = 20:1,
 - 'Property Identifier' = Log_Buffer,
 - 'Result Flags' = {TRUE, TRUE, FALSE},
 - 'Item Count' = 11,
 - 'Item Data' = Records < a, b, c, d, e, f, g, h, i, j, k > in that order.
 - 'First Sequence Number' = 16

[Move clause 9.21.1.10 into BTL Specified Tests and modify]

9.21.1.10 Reading Items by Sequence with Negative Count

Purpose: To verify that the IUT correctly responds to a ReadRange service request to return items specified by indicating a sequence number and the number of items after that sequence to return.

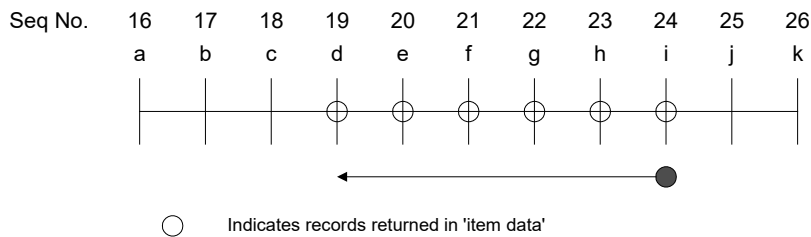
Test Concept: A ReadRange request is transmitted by the TD requesting a range of items known to be in the Log_Buffer. This range is specified using the 'By Sequence' option and a negative value for 'Count'. The 'Reference Sequence Number' and 'Count' are selected so that the results can be conveyed in a single acknowledgment.

Test Steps:

1. TRANSMIT ReadRange-Request,
 - 'Object Identifier' = (the log object configured for this test),
 - 'Property Identifier' = Log_Buffer,
 - 'Reference Sequence Number' = (any value x : $(Total_Record_Count - Record_Count + 2) < x \leq Total_Record_Count$),
 - 'Count' = (any value y : $0 < |y| < (Record_Count - (Total_Record_Count - x) + 1)$)
2. RECEIVE ReadRange-ACK,
 - 'Object Identifier' = (the log object configured for this test),
 - 'Property Identifier' = Log_Buffer,
 - 'Result Flags' = {?, ?, FALSE},
 - 'Item Count' = y ,
 - 'Item Data' = (All of the specified records in order of increasing sequence number. The items specified are all items in the range of $(x - |y| + 1)$ through x in that order.)
 - 'First Sequence Number' = $(x - |y| + 1)$

Test Example (using sample buffer at beginning of section):

1. TRANSMIT ReadRange-Request,
 - 'Object Identifier' = (Trend Log, 1),
 - 'Property Identifier' = Log_Buffer,
 - 'Reference Sequence Number' = 24,
 - 'Count' = -6
2. RECEIVE ReadRange-ACK,
 - 'Object Identifier' = (Trend Log, 1),
 - 'Property Identifier' = Log_Buffer,
 - 'Result Flags' = {FALSE, FALSE, FALSE},
 - 'Item Count' = 6,
 - 'Item Data' = Records < d, e, f, g, h, i > in that order.
 - 'First Sequence Number' = 19



[Delete clause 9.21.1.11 from 135.1-2019]

[This test removal from the standard is proposed as it is no longer referenced by the BTL Test Plan]

[Move test 9.21.1.12 into BTL Specified Tests and move to 7.3.2.24 and modify]

[Replace existing clause with "This clause moved to ..."]

9.21.1.12 7.3.2.24.XI Status/Failure logging

Purpose: To verify that a failure is logged when an error is encountered in an attempt to read a data value from the monitored object. ~~If the error is conveyed by an error response from a remote device, verify that the Error Class and Error Code in the response is logged.~~

Test Concept: ~~Make the monitored object fail and respond with an error by setting the Configure Log_DeviceObjectProperty of the logging object with an unknown object such that collection of records fails. to an invalid device or object.~~ Wait until the IUT attempts to read a sample for the Log_Buffer. Then check the Log_Buffer to verify that there is a failure entry that consists of the ErrorClass and ErrorCode of the error. Repeat with Log_DeviceObjectProperty referencing an object in a device that does not exist.

Configuration Requirements: Configure the logging object so that collection of records will fail (such as by referencing a non-existent object).

Test Steps:

- ~~1. WRITE (Invalid object into the Log_DeviceObjectProperty of the log object)~~
1. MAKE (Log_DeviceObjectProperty reference a non-existent object in the local device or in an existing remote device)
2. WAIT (until IUT attempts to read a sample for the Log_Buffer)
3. ~~VERIFY~~ CHECK(Log_Buffer contains a failure entry of with an error class/error code of
OBJECT/UNKNOWN_OBJECT~~unknown-object~~)
4. IF the IUT supports logging remote values THEN {
 MAKE (Log_DeviceObjectProperty reference an object in a non-existing device)
 WAIT (until IUT attempts to read a sample for the Log_Buffer)
 CHECK (Log_Buffer contains a failure entry with an error class/code
 COMMUNICATION/UNKNOWN_DEVICE)