BACnet® TESTING LABORATORIES ADDENDA

Addendum alm to BTL Test Package 18.1

Revision v3
Revised 8/31/2021

Approved by the BTL Working Group on July 8, 2021.
Approved by the BTL Working Group Voting Members on August 30, 2021.
Published on September 1, 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.

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 ~~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.

BTL-18.1-alm-1: AE-LS-A Testing [BTLWG-973]

Overview:
The AE-LS-A testing currently requires that the IUT intiate LifeSafetyOperation requests both targeted at a single object and targeted at all objects. The BIBB does not mandate the forms that the IUT has to support.

Changes:

Checklist Changes

[Modify section Alarm and Event Management - LifeSafety - A]

Alarm and Event Management - LifeSafety - A		
	R	Base Requirements
	C1	Initiates LifeSafetyOperation requests *targeting a single object*
	C1	*Initiates LifeSafetyOperation requests targeting all life safety objects in a device*
	R	Executes ConfirmedEventNotifications
	R	Executes UnconfirmedEventNotifications
	R	Processes intrinsically generated notifications
	R	Processes algorithmically generated notifications
	R	Processes event notifications with timestamps of the BACnetDateTime form
	R	Processes event notifications with timestamps of the Time form
	R	Processes event notifications with timestamps of the Sequence Number form
	R	Supports AE-ACK-A
	R	Supports AE-INFO-A
[1] *At least one of these options must be supported.*		

Test Plan Changes

[Modify in AE-LS-A, section 5.21.2]

5.21.2 Initiates LifeSafetyOperation Requests *Targeting a Single Object*

135.1-2013 - 8.9.1 - LifeSafetyOperation Service Initiation Tests to an Object		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	
~~**135.1-2013 - 8.9.2 - LifeSafetyOperation Service Initiation Tests to all Objects in a Device**~~		
	~~**Test Conditionality**~~	~~Must be executed.~~
	~~**Test Directives**~~	
	~~**Testing Hints**~~	

[Insert into AE-LS-A, section 5.21.3 -- and renumber all 5.21 sections that follow]

5.21.3 Initiates LifeSafetyOperation Requests Targeting All Life Safety Objects in a Device

135.1-2013 - 8.9.2 - LifeSafetyOperation Service Initiation Tests to all Objects in a Device		
	Test Conditionality	Must be executed.
	Test Directives	
	Testing Hints	

BTL-18.1-alm-2: Mode Transition Test Fix [BTLWG-1016, CR-0477]

Overview:
In test 8.4.8.7, the test requires testing of NORMAL -> NORMAL transitions when Mode changes, but if the IUT only supports one Mode for NORMAL, then the test cannot be passed.

The Test Plan and test should be changed to accommodate the allowable options which an IUT might implement with respect to Mode derived transitions.

Changes:

[Change Clause **BTL Test Plan-18.1.Final - 5.22.11**]

5.22.11 Mode Transition Tests when Event State is Maintained

~~135.1-2013~~*BTL* - 8.4.8.7 - Mode Transition Tests when Event State is Maintained		
	Test Conditionality	~~Must be executed~~ *If the IUT does not support Mode changes which result in the object maintaining its current Event_State, this test shall be skipped.*
	Test Directives	
	Testing Hints	

[Move test **135.1.2019 - 8.4.8.7** to BTL Specified Tests and modify]

8.4.8.7 Mode Transition Tests when Event State is Maintained

Reason for change: Modify the test case as per CR-0477, when IUT does not support Mode changes which maintain the current Event_State.

Purpose: To verify the correct operation of the CHANGE_OF_LIFE_SAFETY event algorithm for objects transitioning between the NORMAL, OFFNORMAL and LIFE_SAFETY_ALARM event states when a mode change occurs. Tests are conducted when a mode change occurs, but the event state does not change. Tests are also conducted when a mode change occurs simultaneously with an event state change. In this latter case, the test verifies that the notification is immediate rather than waiting for the time delay.

Test Concept: The object begins the test in a NORMAL state. The Mode is changed. After the time delay expires, the object should transmit an event notification message. This operation is tested in the OFFNORMAL and LIFE_SAFETY_ALARM states as well.

The test is then repeated by changing the Mode property and simultaneously selecting a pMonitoredValue designated in pAlarmValues. The object should immediately enter the OFFNORMAL state and transmit an event notification message. pMonitoredValue is then changed to a value corresponding to a NORMAL state, and the Mode is simultaneously written. The object should immediately enter the NORMAL state and transmit an event notification message.

Configuration Requirements: The IUT shall be configured such that the Event_Enable property has a value of TRUE for the TO-OFFNORMAL, and TO-NORMAL transitions. The 'Issue_Confirmed_Notifications' parameter shall have a value of TRUE. The event-generating objects shall be in a NORMAL state at the start of the test.

Test Steps:
1. VERIFY Event_Detection_Enable = TRUE
2. CHECK (pCurrentState = NORMAL)
3. MAKE (pMonitoredValue have a value that corresponds to a NORMAL state)
4. *IF (IUT supports another pMode value which maintains the NORMAL state) THEN {*
4. MAKE (pMode = different value that maintains pCurrentState as NORMAL)
5. WAIT (pTimeDelayNormal)
6. BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the intrinsic reporting object being tested or the Event Enrollment object being tested),
'Time Stamp' = (T1: any valid time stamp),
'Notification Class' = (the configured notification class),
'Priority' = (the value configured to correspond to a TO-NORMAL transition),
'Event Type' = CHANGE_OF_LIFE_SAFETY,
'Message Text' = (S1: optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = NORMAL,
'To State' = NORMAL,
'Event Values' = pMonitoredValue, pMode, pStatusFlags, pOperationExpected

~~7~~ TRANSMIT BACnet-SimpleACK-PDU
~~8~~ IF (Protocol_Revision is present AND Protocol_Revision ≥ 13) THEN
VERIFY pStatusFlags = (FALSE, FALSE, ?, ?)
~~9~~ VERIFY pCurrentState = NORMAL
~~10~~ IF (Protocol_Revision is present AND Protocol_Revision ≥ 1) THEN
VERIFY Event_Time_Stamps = (*, *, T1)
~~11~~ IF (Event_Message_Texts property exists) THEN
VERIFY Event_Message_Texts = (*, *, S1)
}
5. ~~12~~MAKE (pMonitoredValue have a value that corresponds to an OFFNORMAL state)
6. VERIFY pCurrentState = OFFNORMAL
7. IF (IUT supports another pMode value which maintains the OFFNORMAL state) THEN {
~~13~~ MAKE (pMode = different value that maintains pCurrentState as OFFNORMAL)
~~14~~ WAIT (pTimeDelay)
~~15~~ BEFORE **Notification Fail Time**
RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the intrinsic reporting object being tested or the Event Enrollment object being tested),
'Time Stamp' = (T2: any valid time stamp),
'Notification Class' = (the configured notification class),
'Priority' = (the value configured to correspond to a TO-OFFNORMAL transition),
'Event Type' = CHANGE_OF_LIFE_SAFETY,
'Message Text' = (S2: optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = OFFNORMAL,
'To State' = OFFNORMAL,
'Event Values' = pMonitoredValue, pMode, pStatusFlags, pOperationExpected
~~16~~ TRANSMIT BACnet-SimpleACK-PDU
~~17~~ IF (Protocol_Revision is present AND Protocol_Revision ≥ 13) THEN
VERIFY pStatusFlags = (TRUE, FALSE, ?, ?)
~~18.~~ VERIFY pCurrentState = OFFNORMAL
~~19.~~ IF (Protocol_Revision is present AND Protocol_Revision ≥ 1) THEN
VERIFY Event_Time_Stamps = (T2, *, *)
~~20.~~ IF (Event_Message_Texts property exists) THEN
VERIFY Event_Message_Texts = (S2, *, *)
}
8. ~~21.~~MAKE (pMonitoredValue have a value that corresponds to a LIFE_SAFETY_ALARM state)
9. *IF (IUT supports another pMode value which maintains the LIFE_SAFETY_ALARM state) THEN {*
~~22.~~ MAKE (pMode = different value that maintains pCurrentState = LIFE_SAFETY_ALARM)
~~23.~~ WAIT (pTimeDelay)
~~24.~~ BEFORE **Notification Fail Time**

RECEIVE ConfirmedEventNotification-Request,
'Process Identifier' = (any valid process ID),
'Initiating Device Identifier' = IUT,
'Event Object Identifier' = (the intrinsic reporting object being tested or the Event Enrollment object being tested),
'Time Stamp' = (T3: any valid time stamp),
'Notification Class' = (the configured notification class),
'Priority' = (the value configured to correspond to a TO-NORMAL transition),
'Event Type' = CHANGE_OF_LIFE_SAFETY,
'Message Text' = (S3: optional, any valid message text),
'Notify Type' = EVENT | ALARM,
'AckRequired' = TRUE | FALSE,
'From State' = OFFNORMAL,
'To State' = OFFNORMAL,
'Event Values' = pMonitoredValue, pMode, pStatusFlags, pOperationExpected

~~25.~~ TRANSMIT BACnet-SimpleACK-PDU

~~26.~~ IF (Protocol_Revision is present AND Protocol_Revision ≥ 13) THEN
VERIFY pStatusFlags = (TRUE, FALSE, ?, ?)

~~27.~~ VERIFY pCurrentState = OFFNORMAL

~~28.~~ IF (Protocol_Revision is present AND Protocol_Revision ≥ 1) THEN
VERIFY Event_Time_Stamps = (T3, *, *)

~~29.~~ IF (Event_Message_Texts property exists) THEN
VERIFY Event_Message_Texts = (S3, *, *)

}