

### Clarification Request

**References:** BTL Test Plan 5.2.10, BTL Specified Tests 8.4.4

**Date of BTL-WG Response:** \_\_07-July-2011\_\_

#### Background:

Attempting to test a device but hedging the test 8.4.4 Test Concept, since decoupling did not seem to be necessary, it was found not that this device does everything correctly when it does issue an event notification of type COMMAND\_FAILURE, but that the test in full cannot be performed, explicitly as written.

BTL - 8.4.4 - COMMAND_FAILURE Tests (ConfirmedEventNotification)		
	<b>Test Method</b>	Manual
	<b>Configuration</b>	As per <i>BTL Specified Tests</i> .
	<b>Test Conditionality</b>	Must be executed. This test must be repeated once for each object type that is capable of generating event notifications with an Event_Type of COMMAND_FAILURE.
	<b>Test Directives</b>	
	<b>Testing Hints</b>	
	<b>Notes &amp; Results</b>	

The following version taken from 135.1-2009I-7

#### 8.4.4 COMMAND\_FAILURE Tests

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

BACnet Reference Clauses: 12.7, 12.12, 12.19, 13.2, 13.3.4, and 13.8.

Purpose: To verify the correct operation of the COMMAND\_FAILURE algorithm. ~~This test applies to Event Enrollment objects with an Event\_Type of COMMAND\_FAILURE and to intrinsic event reporting for Binary Output and Multi State Output objects.~~

Test Concept: The Feedback\_Value (Feedback\_Property\_Reference) shall be decoupled from the input signal that is normally used to verify the output. Initially

Present\_Value (referenced property) and Feedback\_Value (Feedback\_Property\_Reference) are in agreement. Present\_Value (the referenced property) is changed and an event notification should be transmitted indicating a transition to an OFFNORMAL state. The Feedback\_Value (Feedback\_Property\_Reference) is changed to again agree with the Present\_Value (referenced property). A second event notification is transmitted indicating a return to a NORMAL state.

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 property shall have a value of TRUE. The event-generating object shall be in a NORMAL state at the start of the test. The Feedback\_Value property shall be decoupled from the input signal that is normally used to verify the output so that it can be independently manipulated.

In the test description below Present\_Value is used as the referenced property and Feedback\_Value is used to verify the output. If an Event Enrollment object is being tested these properties shall be replaced by the appropriate property reference.

[Note to Reviewers: The modifications to steps 5 and 12 shown below supersede the modifications to these steps shown in section 4 of published Addendum 135.1-2009h.]

#### Test Steps:

1. VERIFY Event\_State = NORMAL
2. IF (the object being tested is not an Event Enrollment object) THEN  
    VERIFY Status\_Flags = (FALSE, FALSE, FALSE, FALSE)
3. IF (Present\_Value is writable) THEN  
    WRITE Present\_Value = (a different value)  
ELSE  
    MAKE (Present\_Value take on a different value)
4. WAIT (Time\_Delay)
5. 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~~  
~~object referenced by the~~ Event Enrollment object being tested),  
        'Time Stamp' = (the current local time),  
        'Notification Class' = (the configured notification class),  
        'Priority' = (the value configured to correspond to a TO-OFFNORMAL transition),  
        'Event Type' = COMMAND\_FAILURE,  
        'Notify Type' = EVENT | ALARM,  
        'AckRequired' = TRUE | FALSE,  
        'From State' = NORMAL,  
        'To State' = OFFNORMAL,  
        'Event Values' = Present\_Value, Status\_Flags, Feedback\_Value
6. TRANSMIT BACnet-SimpleACK-PDU

7. *IF (the object being tested is not an Event Enrollment object) THEN*  
     VERIFY Status\_Flags = (TRUE, FALSE, ?, ?)
8. VERIFY Event\_State = OFFNORMAL
9. *IF (Protocol\_Revision is present and Protocol\_Revision  $\geq$  1) THEN*  
     VERIFY Event\_Time\_Stamps = (the timestamp in step 5, \*, \*)
10. *IF (Feedback\_Value is writable) THEN*  
     WRITE Feedback\_Value = (a value consistent with Present\_Value)  
   ELSE  
     MAKE (Feedback\_Value take on a value consistent with Present\_Value)
11. WAIT (Time\_Delay)
12. **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~~  
         object referenced by the ~~Event Enrollment~~ object being tested),  
         'Time Stamp' = (the current local time),  
         'Notification Class' = (the configured notification class),  
         'Priority' = (the value configured to correspond to a TO-NORMAL transition),  
         'Event Type' = COMMAND\_FAILURE,  
         'Notify Type' = EVENT | ALARM,  
         'AckRequired' = TRUE | FALSE,  
         'From State' = OFFNORMAL,  
         'To State' = NORMAL,  
         'Event Values' = Present\_Value, Status\_Flags, Feedback\_Value
13. TRANSMIT BACnet-SimpleACK-PDU
14. *IF (the object being tested is not an Event Enrollment object) THEN*  
     VERIFY Status\_Flags = (FALSE, FALSE, ?, ?)
15. VERIFY Event\_State = NORMAL
16. *IF (Protocol\_Revision is present and Protocol\_Revision  $\geq$  1) THEN*  
     VERIFY Event\_Time\_Stamps = (the timestamp in step 5, \*, the timestamp in step 12)

Notes to Tester: The 'Message Text' parameter is omitted in the test description because it is optional. The IUT may include this parameter in the notification messages. The time stamps indicated by "\*" in steps 9 and 16 can have a value that indicates an unspecified time or a time that precedes the timestamp in step 5.

When an implementation is seen where Feedback\_Value lags when a write to Present\_Value is executed, momentarily becoming not in agreement, then becoming in agreement again. That indicates "not decoupled from the input signal". They still have some coupling. After making Present\_Value different from Feedback\_Value, we observed two Notifications with Event Type 'Command-Failure': One with 'From State': Normal and 'To State': OffNormal and immediately another Notification with 'From State': OffNormal and 'To State': Normal State. It was observed that IUT then automatically set Feedback value equal to present value. But because of the Notification with 'From State': OffNormal and 'To State': Normal State, we could not execute then steps 7 to 10.

No device is required to exhibit COMMAND\_FAILURE. That is optional on the checklist. That this device does everything correctly when it does issue an event notification of type COMMAND\_FAILURE, is more of a success than a failure. Asking for the implementation to provide more lab accessible configurability than this, for "decoupling from the input signal", purely for test purposes, seems wrong.

An alternate workaround of setting Out\_Of\_Service to TRUE is not in keeping with the concept, explicitly violating step 2, as well as because disagreement between Present\_Value and Feedback\_Value is not then required to generate an Event nor required not to generate an Event.

**Question:**

Should the Test Conditionality of 8.4.4, or revision to Test Plan to allow for an alternate test, be made for a device that can be observed to issue an event notification of type COMMAND\_FAILURE without Feedback\_Value decoupled from the input signal?

**Response:**

The BTL-WG will review the test and test directives to allow the use of this test against a larger set of devices