

Clarification Request

References: ASHRAE 135-2016, BTL Checklist 15.2, BTL Test Plan 15.2

Date of BTL-WG Response: 11-July-2019

☒ All Actions Necessitated have been Completed

Background:

Binary Output Objects that contain certain properties must implement intrinsic reporting.

11 one of the optional properties Elapsed_Active_Time or Time_Of_Active_Time_Reset is
 12 of these properties shall be present.
 13 These properties are required if the object supports intrinsic reporting.
 14 Footnote removed.

In the checklist, the AE-N-I-B has entries for 'Implements intrinsic alarming', 'Implements the CHANGE_OF_RELIABILITY - No Fault Algorithm', 'Implements COMMAND_FAILURE algorithm', etc. None of these are specific to an object.

Problem:

Our Binary Output Object supports intrinsic reporting and generates the CHANGE_OF_RELIABILITY - no fault algorithm. The corresponding entries in the Checklist were selected. The 'Implements COMMAND_FAILULRE algorithm' was not selected on submittal. The lab claims that because the Binary Output Object contains the intrinsic reporting properties, it is required to also support the COMMAND_FAILURE algorithm.

The Binary Output Object cannot actually be configured to generate the COMMAND_FAILURE algorithm because our Feedback_Value implementation does not allow it to be different from the Present_Value.

The following tests are what are being applied for the testing of the COMMAND_FAILURE algorithm: Tests BTL - 8.4.4 COMMAND_FAILURE Tests (ConfirmedEventNotification) and BTL- 8.5.4 COMMAND_FAILURE Tests (UnconfirmedEventNotification).

5.2.11 Implements the **COMMAND_FAILURE** Algorithm¶

The IUT contains, or can be made to contain, an object that can generate ConfirmedEventNotifications and UnconfirmedEventNotifications with an Event_Type of **COMMAND_FAILURE**.¶

BTL-8.4.4- COMMAND_FAILURE Tests (ConfirmedEventNotification)¶		
□	Test Conditionality¶	Must be executed. ¶
□	Test Directives¶	This test must be repeated once for each object type that is capable of generating event notifications with an Event_Type of COMMAND_FAILURE .¶
□	Testing Hints¶	¶
BTL-8.5.4- COMMAND_FAILURE Tests (UnconfirmedEventNotification)¶		
□	Test Conditionality¶	Must be executed. ¶
□	Test Directives¶	This test must be repeated once for each object type that is capable of generating event notifications with an Event_Type of COMMAND_FAILURE .¶
□	Testing Hints¶	¶

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.¶

¶
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.¶

The Checklist and Test Plan do not clearly require the **COMMAND_FAILURE** event algorithm on the Binary Output Object. The requirement is being applied based on the different texts from the standard. See ASHRAE References below.

Question:

Is it required for a Binary Output Object to be able to generate the **COMMAND_FAILURE** algorithm if the Feedback_Value cannot be made different from the Present_Value?

Can the Binary Output Object contain the intrinsic properties if it only supports the **CHANGE_OF_RELIABILITY** algorithm?

Response:

- 1) **No.**
- 2) **Yes.**

ASHRAE References:

See text below from 135-2016:

According to Table 13-5, a Binary Output object is allowed to generate the CHANGE_OF_RELIABILITY algorithm.

Table 13-5. Properties Reported in CHANGE_OF_RELIABILITY Notifications

Object Type	Properties
Access Door	Door_Alarm_State Present_Value
Binary Output, Binary Lighting Output, Multi-state Output	Present_Value Feedback_Value ³

12.7 Binary Output Object Type

The Binary Output object type defines a standardized object whose properties represent the externally visible characteristics of a binary output. A "binary output" is a physical device or hardware output that can be in only one of two distinct states. In this description, those states are referred to as ACTIVE and INACTIVE. A typical use of a binary output is to switch a particular piece of mechanical equipment, such as a fan or pump, on or off. The state ACTIVE corresponds to the situation when the equipment is on or running, and INACTIVE corresponds to the situation when the equipment is off or idle.

In some applications, electronic circuits may reverse the relationship between the application-level logical states, ACTIVE and INACTIVE, and the physical state of the underlying hardware. For example, a normally open relay contact may result in an ACTIVE state (device energized) when the relay is energized, while a normally closed relay contact may result in an INACTIVE state (device energized) when the relay is not energized. The Binary Output object provides for this possibility by including a Polarity property. See Clauses 12.7.4 and 12.7.11.

Binary Output objects that support intrinsic reporting shall apply the COMMAND_FAILURE event algorithm.

12.7.25 Feedback_Value

This property is an indication of the actual value of the entity controlled by Present_Value. The manner by which the Feedback_Value is determined shall be a local matter.

If the object supports event reporting, then this property is the pFeedback parameter for the object's event algorithm. See Clause 13.3 for event algorithm parameter descriptions.

13.3.4 COMMAND_FAILURE Event Algorithm

The COMMAND_FAILURE event algorithm detects whether the monitored value and the feedback value disagree for a time period. It may be used, for example, to verify that a process change has occurred after writing a property.