

**INTERPRETATION IC 135-2010-10 OF
ANSI/ASHRAE STANDARD 135-2010 BACnet® -
A Data Communication Protocol for Building
Automation and Control Networks**

Approval Date: November 7, 2012

Request from: Duffy O'Craven (btl-manager@bacnetinternational.org), Quinda Inc., 41 St. Hilda's Av. Toronto ON M4N 2P5 CANADA.

Reference: This request for interpretation refers to the requirements presented in ANSI/ASHRAE 135-2010, Clauses 12.25 Preamble and 12.27 Preamble and 12.30 Preamble, relating to the behavior of the property Total_Record_Count of logging objects.

Background: This interpretation request originated from the BTL-WG, whilst discussing Clarification Request CR-0274.

The standard never states explicitly when, if ever, the Total_Record_Count, may or shall be reduced by writing to the object.

In Standard 135, in Clause 12.25 preamble states:

The buffer may be cleared by writing a zero to the Record_Count property. Each record in the buffer has an implied SequenceNumber which is equal to the value of the Total_Record_Count property immediately after the record is added.

And in a later provision gives that implied SequenceNumber semantic meaning:

A missed notification may be detected by a subscriber if the 'Current Notification' parameter received in the previous BUFFER_READY notification is different than the 'Previous Notification' parameter of the current BUFFER_READY notification

Similar statements appear in standard 135, in Clause 12.27 and 12.30 preambles.

Also Standard 135.1-2009, in Clause 7.3.2.24.8 with Test concept: "...Record_Count is set to zero and Log_Buffer is read to verify no records are present. Collection of data proceeds until Record_Count is about Buffer_Size/2, collection is halted and Log_Buffer is read to verify the Record_Count value. Collection then resumes until Buffer_Size records are read; collection is then halted and Log_Buffer read to verify Record_Count again." has steps 1-3:

1. WRITE Record_Count = 0
2. WAIT Internal Processing Fail Time
3. CHECK (that Log_Buffer has no records)

Test 7.3.2.24.8 is executed if and only if the IUT is protocol revision 3 or higher.

There are other tests which also write a 0 to Record_Count, but never state any constraint upon Total_Record_Count to give guidance. For instance in test 7.3.2.24.1:

1. WRITE Log_Enable = FALSE
2. WRITE Record_Count = 0

3. WAIT Internal Processing Fail Time
4. TRANSMIT ReadProperty-Request,
 - 'Object Identifier' = (the object being tested),
 - 'Property Identifier' = Total_Record_Count
5. RECEIVE ReadProperty-ACK,
 - 'Object Identifier' = (the object being tested),
 - 'Property Identifier' = Total_Record_Count
 - 'Property Value' = (any valid value, X)

Interpretation: Though writes of 0 to Record_Count property in logging objects clear the buffer, and writes of new values to Log_DeviceObjectProperty render the contents of the buffer obsolete and may clear the buffer, writing to either of those properties shall not reduce Total_Record_Count, except in the condition where incrementing Total_Record_Count wraps around its maximum possible value of $2^{32} - 1$, as its next value then becomes 1.

Question: Is this interpretation correct?

Answer: Yes.

Comments: It does not matter how the buffer is cleared, be it by writing 0 to Record_Count or changing Log_DeviceObjectProperty, the Total_Record_Count does not reset to 0.