ERRATA SHEET FOR

ASHRAE/IESNA STANDARD135-95, BACnet® - A Data Communications Protocol for Building Automation and Control Networks

September 7, 1999

This document lists all known *errata* to ANSI/ASHRAE 135-1995 as of the above date. Each entry is cited first by clause, then page number.

1) 5.1, p. 15: There are actually no ERROR.REQ and ERROR.IND primitives defined in the standard. Errors are conveyed by CONFIRMED_SERV. response(-), which generates a BACnet-Error-PDU. This error was introduced during the last-minute editing process. Therefore, correct the third and fourth paragraphs on p. 15 as follows:

CONF_SERV.request CONF_SERV.indication CONF_SERV.response CONF_SERV.confirm
UNCONF_SERV.request UNCONF_SERV.indication
SEGMENT_ACK.request SEGMENT_ACK.indication
~~ERROR.request ERROR.indication~~
REJECT.request REJECT.indication
ABORT.request ABORT.indication

The designation CONF_SERV indicates that BACnet confirmed service PDUs are being used. Similarly, the designations UNCONF_SERV, SEGMENT_ACK, ~~ERROR,~~ REJECT, and ABORT indicate that unconfirmed service PDUs, segment acknowledge PDUs, ~~error PDUs,~~ reject PDUs, and abort PDUs, respectively, are being used.

2) 5.4.4.3, p. 30: State Machine for Requesting BACnet User (client) has an error which occurs if a TimeoutUnsegmented event occurs while in the AWAIT_CONFIRMATION state. It re-sends the PDU and again awaits confirmation, but does not restart the RequestTimer.

Therefore, in the action portion of the TimeoutUnsegmented transition, change:

...and enter the AWAIT_CONFIRMATION state to await a reply.

to:

...start RequestTimer; and enter the AWAIT_CONFIRMATION state to await a reply.

3) 5.4.4.3, p. 30: State Machine for Requesting BACnet User (client) leaves SentAllSegments = TRUE if a TimeoutSegmented event occurs while waiting for confirmation. This can cause a problem in the subsequent state SEGMENTED_REQUEST if the tests "...there is at least one segment remaining to send" (in the NewACK_Received predicate) and "...there are no more segments to send" (in the FinalACK_Received predicate) are based upon SentAllSegments.

Therefore, in the action portion of the TimeoutSegmented transition, change:

...set SegmentRetryCount to zero;...

to:

...set SegmentRetryCount to zero; *set SentAllSegments to FALSE;...*

4) 5.4.4.3, p. 30: State Machine for Requesting BACnet User (client) retries one time too many because of an error in the FinalTimeout event.

To correct this counting error, in the predicate of FinalTimeout change:

...and RetryCount is greater than Number_Of_APDU_Retries,

to:

...and RetryCount is greater than *or equal to* Number_Of_APDU_Retries,

5) 5.4.5, p. 32ff: State Machine for Responding BACnet User (server) has a Timeout event in the AWAIT_RESPONSE state, p. 36, based on the expiration of RequestTimer but the timer is never set.

To correct this, change 5.4.5.1 IDLE, ConfirmedUnsegmentedReceived, p. 32:

...program, and enter the AWAIT_RESPONSE state.

to:

...program, *start RequestTimer,* and enter the AWAIT_RESPONSE state.

Also change 5.4.5.2 SEGMENTED_REQUEST, LastSegmentOfMessageReceived, p. 34:

...program; and enter the AWAIT_RESPONSE state.

to:

...program; *start RequestTimer;* and enter the AWAIT_RESPONSE state.

6) 6.2.2, p. 52: add closing ")" to DLEN=...(...

7) 6.2.2.1, p. 54: Add the text shown in *italics* so that the clause reads as follows.

The multi-octet fields, DNET, SNET, and Vendor ID, shall be conveyed with the most significant octet first. *Allowable network number values for DNET shall be from 1 to 65535 and for SNET from 1 to 65534.*

8) 6.7, Figure 6-13, p. 68: In the caption, change "half routers" to "half-routers".

9) 9.5.6, Figure 9-4, p. 91: Add the *SoleMasterRestartMaintenancePFM* transition from the DONE_WITH_TOKEN to the POLL_FOR_MASTER state. See item 11) below.

10) 9.5.6.2, p. 92: Change the ReceivedToken action to:

then set ReceivedValidFrame to FALSE; set FrameCount to zero; *set SoleMaster to FALSE;* and enter the USE_TOKEN state.

11) 9.5.6.5, p. 95: Change the ResetMaintenancePFM test as shown.

If FrameCount is greater than or equal to Nmax_info_frames, TokenCount is greater than or equal to Npoll, (PS+1) modulo (N_{max_master}+1) is equal to NS, *and SoleMaster is FALSE,*

12) 9.5.6.5, p. 95: Following ResetMaintenancePFM, add this new test and action called *SoleMasterRestartMaintenancePFM*.

If FrameCount is greater than or equal to Nmax_info_frames, TokenCount is greater than or equal to Npoll, (PS+1) modulo (Nmax_master+1) is equal to NS, and SoleMaster is TRUE,

then set PS to (NS+1) modulo (Nmax_master+1); call SendFrame to transmit a Poll For Master to PS; set NS to TS (no known successor node); set RetryCount, TokenCount, and EventCount to zero; and enter the POLL_FOR_MASTER state to find a new successor to TS.

13) 9.5.6.8, p. 96: Change the ReceivedReplyToPFM action as shown.

then *set SoleMaster to FALSE;* set NS equal to SourceAddress; set EventCount to zero; call SendFrame to transmit a Token frame to NS; set PS to the value of TS; set TokenCount and RetryCount to zero; set ReceivedValidFrame to FALSE; and enter the PASS_TOKEN state.

14) 10.4.9.2, p. 122: 10.4.9.2, p.122: To make the "then" clause of the ConnectRequestFailure transition consistent with the language elsewhere in Clause 10 make the following change:

then set RetryCount to RetryCount + 1; retransmit the "BACnet<CR>" trigger ~~phrase~~ *sequence*, ...

15) 10.4.9.3, p. 123: Correct the InvalidConnectResponseReceived then clause to read:

then set ReceivedValidFrame to FALSE; call SendFrame to transmit a Disconnect Request frame indicating the receipt of an invalid password; *set ResponseTimer to zero; set RetryCount to zero;* and enter the DISCONNECTING state.

16) 10.4.9.4, p. 123: Correct the NetworkDisconnect then clause to read:

then call SendFrame to transmit a Disconnect Request frame; set ResponseTimer to zero; issue a DL-DISCONNECT.confirm to notify the network layer of the disconnection; *set RetryCount to zero;* and enter the DISCONNECTING state.

17) 10.4.10.1, p. 124-5: In the ConnectionEstablishedXON and ConnectionEstablishedXOFF transitions, correct the then clauses to read:

...set TxSequenceNumber to zero; *set HeartbeatTimer to zero;* and enter the TRANSMIT BLOCKED state.

18) 10.4.10.2, p. 126: In the HeartbeatTimerExpiredXON and HeartbeatTimerExpiredXOFF transitions, correct the then clauses to read:

...frame*; set HeartbeatTimer to zero*; and enter the TRANSMIT BLOCKED state.

19) 10.4.10.3, p. 126-7: In the HeartbeatTimerExpiredXON and HeartbeatTimerExpiredXOFF transitions, correct the then clauses to read:

...frame*; set HeartbeatTimer to zero*; and enter the TRANSMIT READY state.

20) 12.1.8, p. 139; 12.2.8, p. 144; 12.3.7, p. 149; 12.4.8, p. 155; 12.5.8, p. 161; 12.6.7, p. 167; and 12.13.7, p. 193: It has been pointed out that there are two disconnected uses for "FAULT" in the intrinsic-reporting objects. One use defines a bit in the Status_Flags property; the value of the FAULT bit is a function of the (optional) Reliability property. The other use is in the two properties associated with intrinsic reporting, Event_Enable and Acked_Transitions, which convey flags related to TO-FAULT events. Nowhere are the two usages related, nor is the "fault" enumeration of the Event_State property referenced.

The following text links the two usages for intrinsic-reporting objects and is to be appended, in each clause cited above, to the description of the Event_State property.

If the Reliability property is present and does not have a value of NO_FAULT_DETECTED, then the value of the Event_State property shall be FAULT. Changes in the Event_State property to the value FAULT are considered to be "fault" events.

21) 12.2.10, p. 145 and 12.5.10, p. 161: There is an inconsistency in the language that describes what is to happen when Out_Of_Service is TRUE for Analog Output and Binary Output object types. The intent was that the Reliability and Status_Flags would be decoupled from the physical process to allow the values to be changed for testing purposes (to verify control algorithms, not conformance to BACnet).

In both clauses, add the text shown in *italics* so that the clause reads as follows.

...In addition, the Reliability property and the corresponding state of the FAULT flag of the Status_Flags property shall be decoupled *from the physical output* when Out_Of_Service is TRUE...

22) 12.9.32, p. 180: Add the text shown in *italics* so that the clause reads as follows.

The Device_Address_Binding property is a List of BACnetAddressBinding each of which consists of a BACnet Object_Identifier of a BACnet Device object and a BACnet device address in the form of a BACnetAddress. Entries in the list identify the actual device addresses that will be used when the remote device must be accessed via a BACnet service request. *A value of zero shall be used for the network-number portion of BACnetAddress entries for other devices residing on the same network as this device.* The list may be empty if no device identifier-device address bindings are currently known to the device.

23) 12.13, p. 190: In Table 12-16 the Property Datatype of Event_Enable is misspelled. It should be "BACnetEventTr*a*nsitionBits".

24) 12.18.4, p. 213: Add language to the end of the clause to clarify the value of the Present_Value property of the Schedule object when the List_Of_Object_Property_References is empty:

If the List_Of_Object_Property_References is empty, then the value of this property will be that which would have been most recently written to the List_Of_Object_Property_References.

25) 13, p. 216, third paragraph: the first sentence should be "Intrinsic *reporting* allows..." not "Intrinsic allows..."

26) 13.3, p. 222: The following insertion clarifies an ambiguity in the Standard, stating that implementations of Algorithmic Change Reporting (i.e., by means of Event Enrollment objects) shall be able to issue notifications for transitions to and from the FAULT state as well as NORMAL and OFF-NORMAL, and specifies the use of the ‘status-flags’ notification parameter in algorithmic reporting:

...be forced to return a list of property values without being able to specify the object type of the properties returned.

If the referenced object and property is any of those appearing in Table 13-3, then the value of the Status_Flags property of the referenced object shall be conveyed by the 'status-flags' parameter of the ConfirmedEventNotification or UnconfirmedEventNotification service request issued by the Event Enrollment object. A change in the FAULT flag (independent of any Time_Delay notification parameter) of the referenced object's Status_Flags property shall be treated as if the referenced object's Event_State property had made the associated transition to or from the FAULT state and a notification issued if notification for the resultant transition is enabled.

If the referenced object does not have a Status_Flags property then the 'status-flags' parameter shall convey the values {FALSE, FALSE, FALSE, FALSE} in any ConfirmedEventNotification or UnconfirmedEventNotification service request issued by the Event Enrollment object.

When an Event Enrollment object is created, its Event_State property shall be initialized to NORMAL....

27) Figure 13-9, p. 230: In the Notification Class Object block, the Priority property should be a BACnetARRAY [3] of Unsigned. Thus the integer "3" should be replaced by "3, 3, 3".

28) 13.9.1, p. 240: In Table 13-9, the Notification Class parameter in the Result(+) is incorrectly shown as mandatory, "M M(=)". This should be corrected to read "U U(=)".

29) 13.10.1.5, p. 244: Change last sentence to read "If the 'Lifetime' parameter is present then the 'Issue Confirmed Notifications' parameter shall be present." This makes it consistent with the service procedure of 13.10.2.

30) 13.11.1, p. 246: The title of Table 13-11 should read: "Structure of UnconfirmedCOVNotification Service Primitive".

31) 15.9.2, p. 279: The first two paragraphs of the WritePropertyMultiple service procedure should be combined into a single paragraph and read:

For each 'Write Access Specification' contained in the 'List of Write Access Specifications'~~ the responding BACnet-user shall~~

~~verify that the object specified is present in the local device, that each referenced property is available for modification, and that each specified property value is of the correct data type and within the correct range. If the verification attempt is successful~~, the value of each specified property shall be replaced by the property value provided in the 'Write Access Specification' and a 'Result(+)' primitive shall be issued, indicating that the service request was carried out in its entirety. Intrepretation of the conditional Priority parameter shall be as specified in Clause 19.

32) 20.2.9, p. 340: In the second example on the page, change "Misrosoft" to "Microsoft".

33) 20.2.17, p. 346: In the third example, the application tags should be 10 for Date and 11 for Time yielding encoded tags of X'A4' and X'B4' respectively.

34) 20.2.18, p. 347: In the second example, the application tags should be 10 for Date and 11 for Time yielding encoded tags of X'A4' and X'B4' respectively.

35) 21, p. 365: Add the following comment shown in *italics*.

BACnetAddress ::= SEQUENCE {

network-number Unsigned16, -- *A value of 0 indicates the local network*

mac-address OCTET STRING -- A string of length 0 indicates a broadcast

}

36) 21, p. 374: In the BACnet ObjectType and BACnetObjectTypesSupported productions, the entries for "multistate-input" and "multistate-output" should be changed to "multi-state-input" and multi-state-output" to be consistent with the other places in the standard where we use a hyphen.

37) 21, p. 374: BACnetPriorityArray should be SIZE (16) <u>not</u> SIZE (1..16) because the array is fixed length.

38) 21, p. 378: The choice of Unsigned was omitted from the BACnetPropertyStates production. This is needed to be able to apply the CHANGE_OF_STATE algorithm referenced by an

Event Enrollment object to Multi-state Input and Output objects. It should read:

BACnetPropertyStates ::=
CHOICE {

.

unsigned-value [11] Unsigned,

...

}

39) 21, p. 378: The comment field below the BACnetPropertyStates production should read in part: "... Tag values of 64-*254* may be used by others..."

40) 21 , p. 381: In the BACnetTimeValue production, add a comment after ABSTRACT-SYNTAX&Type that states "-- any primitive datatype". The problem here is that it is impossible to decode any complex datatype without having the abstract syntax expanded with some other known syntax. In reality this is not going to happen so we should warn people. This question has come up from people trying to implement this.

41) 21 , p. 381: In the BACnetWeekNDay production, change the last item in the second octet description to read "*X'FF'* = any week of this month" for consistency with the first and third octet descriptions.

42) 23.1, p. 391: In Table 23-1, the Maximum Value for BACnetPropertyStates should be 254.

43) D.3, p. 420: The units should be BTUS-PER-POUND-DRY-AIR (plural of BTU) in order to match the ASN.1 production on p. 367.

44) D.7, p. 423: The property "DateList" should be "Date_List". Also the "=" sign is missing following the property name.

45) D.9, p. 425, 426: These changes pertain to Example 1:

The value of Max_APDU_Length_Accepted should be changed to 480 (see p. 324).

The value "VT100" in VT_Classes_Supported should be "DEC-VT100".

Remove the quotation marks around the value for Local_Time. This is not a string.

Remove the quotation marks around the value for Local_Date. This is not a string. We should also add "Friday" to make it consistent with the format of a BACnet date. It should read: 29-SEP-1989, FRIDAY.

The APDU timeout value is currently 60,000. This is consistent with the standard, but not with real life. We should use a more realistic example, like 3,000.

In the List_of_Session_Keys, the value "3)" in line 1 should be "X'03')". In line 2 the "5)" should be "X'05')".

In the Device_Address_Binding value, line 1 - the final "1" should be X'01'. In line 2 there should be an additional opening parenthesis at the left margin and the "23" at the end should be "X'23'". In line 3 there should be an additional opening parenthesis at the left margin.

These changes pertain to Example 2: In the Device_Address_Binding value, line 1 - the final "1" should be X'01'. In the second line, on p. 426, there should be an additional opening parenthesis at the left margin and the "23" at the end should be "X'23'".

46) D.11, p. 427: On the last line of the page, File_Access_Method should be followed by an equals sign, '='.

47) D.14, p. 429, 430: In both examples, the Object_Type should be MULTI_STATE_INPUT, (the first underscore is missing). On p. 430 there should be parentheses around the "3" in Alarm_Values and around the "2" in Fault_Values since these are lists.

48) D.15, p. 430, 431: In both examples, the Object_Type should be MULTI_STATE_OUTPUT, (the first underscore is missing).

49) I, p. 499: The following editorial clarification clarifies the significance of the priority array levels.

"Because the state of the Present_Value has changed, the minimum on and off time maintenance entity writes the new state of

ACTIVE to entry 6 in the Priority_Array in order to enforce the minimum time. Any further writes to the Present_Value at priorities ~~less than 6~~ *numerically greater than 6 (less important)* will be entered into the Priority_Array but will not be acted upon due to the presence of the ACTIVE at priority 6. For example, in Figure I-1(c), a write to Present Value at priority 7 with a value of INACTIVE will be entered into the Priority_Array but will not be acted upon.

Writes to priorities ~~higher than 6~~ *numerically less than 6* will be entered into the Priority_Array and may cause changes of state due to their higher priority. This is desired for emergency and fire control.

In Figure I-1(e) the minimum on and off time maintenance entity in the device issues a relinquish (NULL) at priority 6 when the Minimum_On_Time expires, 10 minutes after the initial write. The remaining entries in the Priority_Array are then examined to determine the new Present_Value. If no change of state results, then no further action occurs. This would be the case in our example if no INACTIVE requests ~~above priority 9~~ *at priorities numerically less than 9* had been made.

We note that while write to priorities ~~above~~ *numerically less than* 6 are not subject to minimum on and off times *(see 19.3)*, if such writes cause changes of states the server should still write the new value to priority 6. Thus, if the higher priority request is relinquished within the minimum time, the minimum will be enforced before any lower priority requests can cause changes of state."