BACnet *Errata*
ANSI/ASHRAE STANDARD 135-2012
A Data Communication Protocol for Building Automation and Control Networks

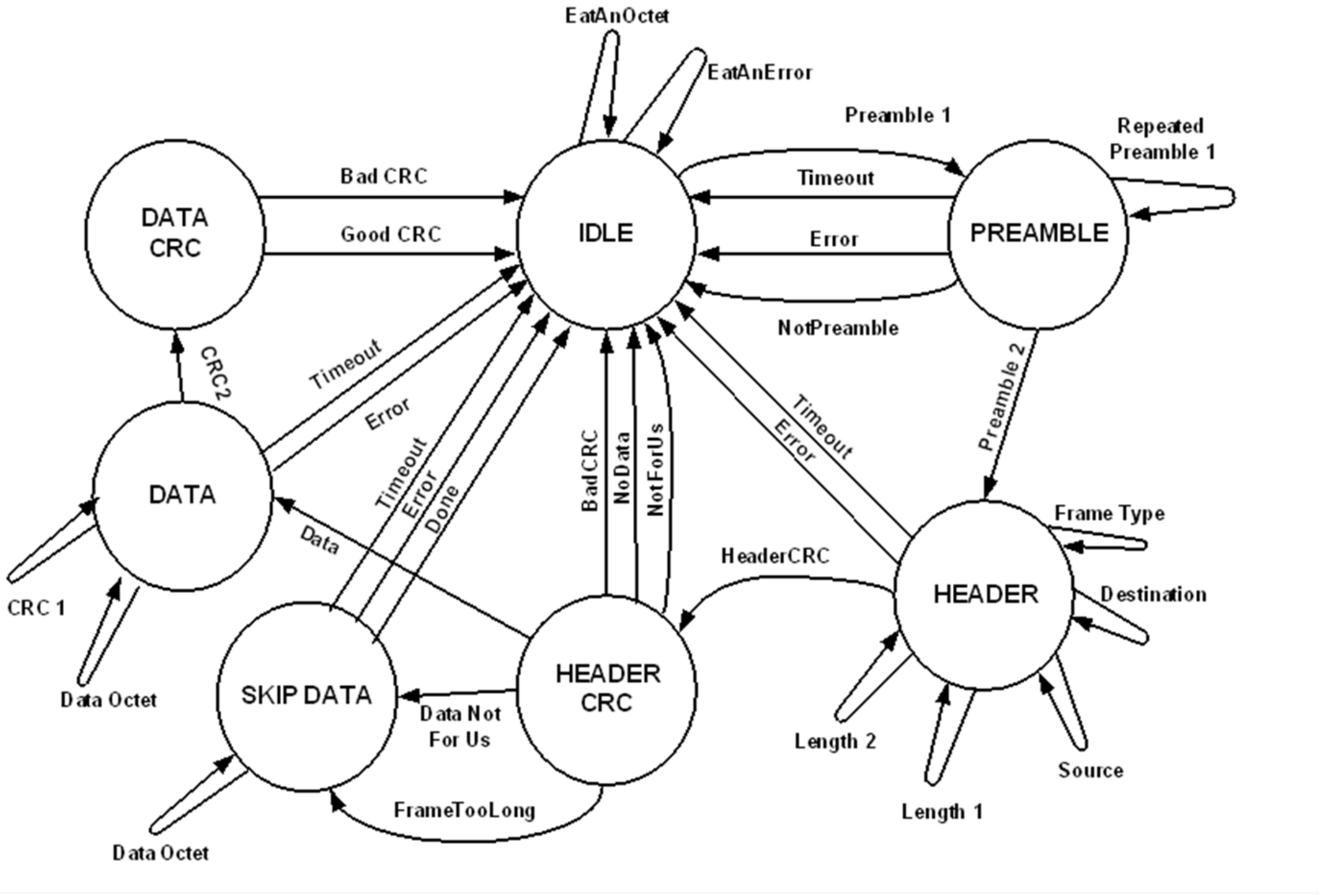
June 29, 2015

This document lists all known ***errata*** to ANSI/ASHRAE 135-2012 as of the above date. Each entry is cited first by clause, then page number, except where an erratum covers more than one clause. The outside back cover marking identifying the first printing of Standard 135-2012 is "Product Code: 86439 3/13".

Changes are indicated by using ~~strikeout~~ for text to be removed and ***italics*** for text to be added, unless noted otherwise.

1) Figure 9-3 is not readable.

[Replace **Figure 9-3**, p. 97]



2) Some error codes are not shown in bold.

[Change **Clauses 18.1** through **18.10**, p. 595]
[Change all error code names to be bold as per the below example]

~~CONFIGURATION_IN_PROGRESS~~**CONFIGURATION_IN_PROGRESS -** A service request has been temporarily declined because the addressed BACnet device is in the process of being configured, either by means local to the device or by means of other protocol services.

3) The UNSIGNED_RANGE event algorithm does not take into account the Limit_Enable property which is present in object types that use the algorithm.

[Change **Clause 13.3.9,** p. 489]

13.3.9 UNSIGNED_RANGE Event Algorithm

The UNSIGNED_RANGE event algorithm detects whether the monitored value exceeds a range defined by a high limit and a low limit.

The parameters of this event algorithm are:

...

pHighLimit This parameter, of type Unsigned, represents the higher limit of the range considered normal.

pLimitEnable *This parameter, of type BACnetLimitEnable, represents two flags, HighLimitEnable and LowLimitEnable, that separately enable (TRUE) or disable (FALSE) the respective limits applied by the event algorithm.*
If the value of this parameter is not provided, then both flags shall be set to TRUE (1).

pTimeDelay This parameter, of type Unsigned, represents the time, in seconds, that the offnormal conditions must exist before an offnormal event state is indicated.

pTimeDelayNormal This parameter, of type Unsigned, represents the time, in seconds, that the Normal conditions must exist before a NORMAL event state is indicated. If no value is provided for this parameter, then it takes on the value of the pTimeDelay parameter.

The conditions evaluated by this event algorithm are:

(a) If pCurrentState is NORMAL, *and the HighLimitEnable flag of pLimitEnable is TRUE,* and pMonitoredValue is greater than pHighLimit for pTimeDelay, then indicate a transition to the HIGH_LIMIT event state.

(b) If pCurrentState is NORMAL, *and the LowLimitEnable flag of pLimitEnable is TRUE,* and pMonitoredValue is less than pLowLimit for pTimeDelay, then indicate a transition to the LOW_LIMIT event state.

(c) If pCurrentState is HIGH_LIMIT, and the HighLimitEnable flag of pLimitEnable is FALSE, then indicate a transition to the NORMAL event state.

~~(c)~~*(d)* Optional: If pCurrentState is HIGH_LIMIT, *and the LowLimitEnable flag of pLimitEnable is TRUE,* and pMonitoredValue is less than pLowLimit for pTimeDelay, then indicate a transition to the LOW_LIMIT event state.

~~(d)~~*(e)* If pCurrentState is HIGH_LIMIT, and pMonitoredValue is equal to or less than pHighLimit for pTimeDelayNormal, then indicate a transition to the NORMAL event state.

(f) If pCurrentState is LOW_LIMIT, and the LowLimitEnable flag of pLimitEnable is FALSE, then indicate a transition to the NORMAL event state.

~~(e)~~*(g)* Optional: If pCurrentState is LOW_LIMIT, *and the HighLimitEnable flag of pLimitEnable is TRUE,* and pMonitoredValue is greater than pHighLimit for pTimeDelay, then indicate a transition to the HIGH_LIMIT event state.

~~(f)~~*(h)* If pCurrentState is LOW_LIMIT, and pMonitoredValue is equal to or greater than pLowLimit, for pTimeDelayNormal, then indicate a transition to the NORMAL event state.

If any of the optional conditions are supported, then all optional conditions shall be supported.

...

[Change **Figure 13-7**, p. 491]

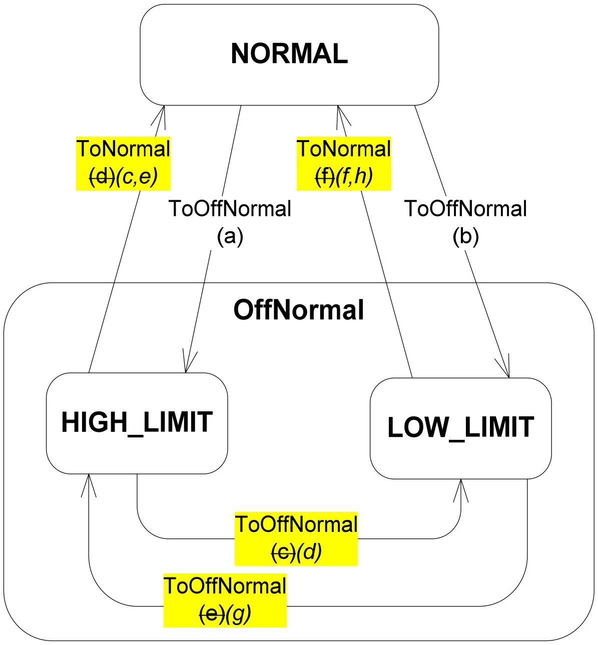
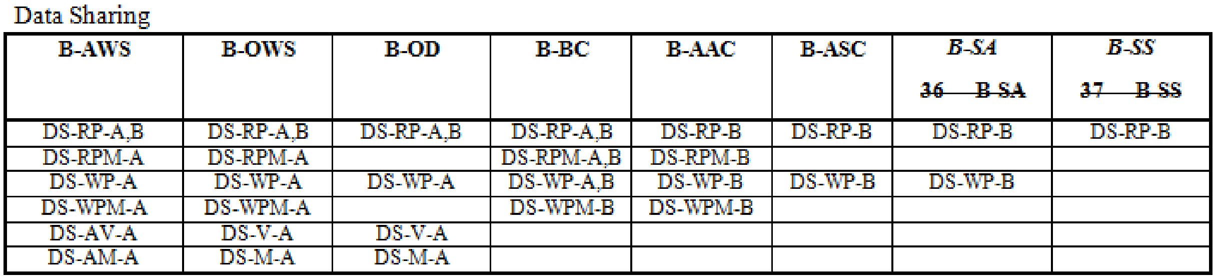


Figure 13-17. Transitions indicated by UNSIGNED_RANGE algorithm

4) The Data Sharing table in L.7 has spurious numbers in the header row.

[Change **Clause L.7**, p. 910]

…



Data Sharing

B-AWS	**B-OWS**	**B-OD**	**B-BC**	**B-AAC**	**B-ASC**	***B-SA*** ~~**36 B-SA**~~	***B-SS*** ~~**37 B-SS**~~
DS-RP-A,B	DS-RP-A,B	DS-RP-A,B	DS-RP-A,B	DS-RP-B	DS-RP-B	DS-RP-B	DS-RP-B
DS-RPM-A	DS-RPM-A		DS-RPM-A,B	DS-RPM-B			
DS-WP-A	DS-WP-A	DS-WP-A	DS-WP-A,B	DS-WP-B	DS-WP-B	DS-WP-B	
DS-WPM-A	DS-WPM-A		DS-WPM-B	DS-WPM-B			
DS-AV-A	DS-V-A	DS-V-A					
DS-AM-A	DS-M-A	DS-M-A					

…

5) The property descriptions for a number of alarming properties are missing for the Channel Object.

[Insert before **Clause 12.53.19**, p. 446]

12.53.19 Event_Detection_Enable

This property, of type BOOLEAN, indicates whether (TRUE) or not (FALSE) intrinsic reporting is enabled in the object and controls whether (TRUE) or not (FALSE) the object will be considered by event summarization services.

This property is expected to be set during system configuration and is not expected to change dynamically.

When this property is FALSE, Event_State shall be NORMAL, and the properties Acked_Transitions, Event_Time_Stamps, and Event_Message_Texts shall be equal to their respective initial conditions.

12.53.20 Notification_Class

This property, of type Unsigned, shall specify the instance of the Notification Class object to use for event-notification-distribution.

12.53.21 Event_Enable

This property, of type BACnetEventTransitionBits, shall convey three flags that separately enable and disable the distribution of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL notifications (see Clause 13.2.5). A device is allowed to restrict the set of supported values for this property but shall support (T, T, T) at a minimum.

12.53.22 Event_State

The Event_State property, of type BACnetEventState, is included in order to provide a way to determine whether this object has an active event state associated with it (see Clause 13.2.2.1). If the object supports event reporting, then the Event_State property shall indicate the event state of the object. If the object does not support event reporting then the value of this property shall be NORMAL.

12.53.23 Acked_Transitions

This read-only property, of type BACnetEventTransitionBits, shall convey three flags that separately indicate the acknowledgment state for TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events (see Clause 13.2.2.1.5). Each flag shall have the value TRUE if no event of that type has ever occurred for the object.

12.53.24 Notify_Type

This property, of type BACnetNotifyType, shall convey whether the notifications generated by the object should be Events or Alarms. The value of the property is used as the value of the 'Notify Type' service parameter in event notifications generated by the object.

12.53.25 Event_Time_Stamps

This read-only property, of type BACnetARRAY[3] of BACnetTimeStamp, shall convey the times of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events (see Clause 13.2.2.1). Timestamps of type Time or Date shall have X'FF' in each octet and Sequence Number timestamps shall have the value 0 if no event of that type has ever occurred for the object.

12.53.26 Event_Message_Texts

This read-only property, of type BACnetARRAY[3] of CharacterString, shall convey the message text values of the last TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events (see Clause 13.2.2.1). If a particular type of event has yet to occur, an empty string shall be stored in the respective array element.

12.53.27 Event_Message_Texts_Config

This property, of type BACnetARRAY[3] of CharacterString, contains the character strings which are the basis for the 'Message Text' parameter for the event notifications of TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively, generated by this object. The character strings may optionally contain proprietary text substitution codes to incorporate dynamic information such as date and time or other information.

[Renumber **Clause 12.53.19**, p. 446]

12.53~~.19~~*.28* Profile_Name

…

6) Incorrectly formatted enumeration ASN.1 constructs in Clause 21 (missing () around values.)

[Change **Clause 21**, p. 713]

BACnetWriteStatus ::= ENUMERATED {
idle *(*0*)*,
in-progress *(*1*)*,
successful *(*2*)*,
failed *(*3*)*
}

7) Incorrect case in SEQUENCE OF expression.

[Change **Clause 21**, p. 651]

WriteGroup-Request ::= SEQUENCE {
groupNumber [0] Unsigned32,
writePriority [1] Unsigned (1..16),
changeList [2] SEQUENCE ~~of~~*OF* BACnetGroupChannelValue,
inhibitDelay [3] BOOLEAN OPTIONAL
}

8) ASN.1 Comments are missing the ASN.1 comment prefix '-- '.

[Insert space after '--' in **Clause 21**, p. 639]

-- ******************* APDU Definitions *******************

[Insert '-- ' before each incorrect comment line in **Clause 21**, p. 639]

-- ********************** Confirmed Service Productions *********************

-- *********************** Confirmed Alarm and Event Services ******************

-- ************************** Confirmed File Access Services *************************

-- ************************** Confirmed Object Access Services ***********************

-- ************************ Confirmed Remote Device Management Services *****************

-- ********************** Confirmed Virtual Terminal Services ********************

-- ******************* Unconfirmed Request Productions ********************

-- ******************* Unconfirmed Alarm and Event Services ********************

-- ******************* Unconfirmed Object Access Services ********************

-- ******************* Unconfirmed Remote Device Management Services ********************

-- ****************************** Error Productions *************************

9) Missing comma in ASN.1

[Change **Clause 21**, p. 708]

lighting-transition [40] BACnetLightingTransition,

10) The Global Group object was missed when indicating which properties are reported in CHANGE_OF_RELIABILITY notifications.

[Change **Table 13-5**, p. 473]

Table 13-5. Properties Reported in CHANGE_OF_RELIABILITY Notifications

Object Type	Properties
…	…
Analog Input, Analog Output, Analog Value, Binary Input, Binary Value, BitString Value, CharacterString Value, ***Global Group,*** Integer Value, Large Analog Value, Multi-state Input, Multi-state Value, Positive Integer Value, Pulse Converter	Present_Value
…	…

11) The requirement for building controllers to be able to command half-routers to connect and disconnect PTP links was removed from the table of required BIBBs in Protocol_Revision 5 but the corresponding text in the building controller description was not removed.

[Change **Annex L.2**, p. 907]

L.2 BACnet Building Controller (B-BC)

...

Device and Network Management

- Ability to respond to queries about its status
- Ability to respond to requests for information about any of its objects
- Ability to respond to communication control messages
- Ability to synchronize its internal clock upon request
- Ability to perform re-initialization upon request
- Ability to upload its configuration and allow it to be subsequently restored
- ~~Ability to command half-routers to establish and terminate connections~~

12) The Loop object incorrect identifies Present_Value and Controlled_Variable_Value as the pMonitoredValue parameters for intrinsic reporting. The correct pMonitoredValue is the Controlled_Variable_Value property.

12.17.4 Present_Value

This property indicates the current output value of the loop algorithm in units of the Output_Units property. The Present_Value property shall be writable when Out_Of_Service is TRUE.

~~If the object supports event reporting, then this property shall be the pMonitoredValue parameter for the object's event algorithm. See Clause 13.3 for event algorithm parameter descriptions.~~

13) Grammatical error in 12.16.12.

[Change **Clause 12.16.12**, p. 230]

12.16.12 Mode

This writable property, of type BACnetLifeSafetyMode, shall convey the desired operating mode for the object.

If the object supports event reporting, then this property shall be the pMode ***parameter***~~parameters~~ of the object's event algorithm. See Clause 13.3 for event algorithm parameter descriptions.

14) The last sentence of clause 6.3 conflicts with the allowance for the broadcasting of confirmed requests described in clause 6.5.3 when the address of the first intervening router is unknown.

[Change **Clause 6.3**, p. 60]

6.3 Messages for Multiple Recipients

BACnet supports the transmission of messages to multiple recipients through the use of multicast and broadcast addresses. Multicasting results in a message being processed by a group of recipients. Broadcasting results in a message being processed by all of the BACnet Devices on the local network, a remote network, or all networks. The use of broadcast or multicast addressing for network layer protocol messages is described in 6.5. Of the BACnet APDUs, only the BACnet-Unconfirmed-Request-PDU may be transmitted using a multicast or broadcast ***network layer*** address ***(note that a MAC layer multicast or broadcast address may be used for other PDU types when the network layer address restricts the destination to a single device)***.

15) The BACnetPropertyIdentifier for the deprecated property List_Of_Session_Keys is not documented in the same manner as other deprecated properties.

[Change **Clause 21**, p. 702]

...

-- see list-of-object-property-references (54),

~~-- enumeration value 55 is unassigned~~

-- formerly: list-of-session-keys (55), removed in version 1 revision 11.

-- see local-date (56),

...

16) Typo in BACnetProcessIdSelection production.

[Change **Clause 21**, p. 693]

BACnetProcessIdSelection ::= CHOICE {

processIdentifier Unsigned32,

~~nullValu~~*nullValue* NULL

}

17) In the UNSIGNED_OUT_OF_RANGE Event Algorithm description, there is a blank line included in the list of conditions evaluated by the algorithm.

[Change **Clause 13.3.5**, p. 500]

...

The conditions evaluated by this event algorithm are:

(a) If pCurrentState is NORMAL, and the HighLimitEnable flag of pLimitEnable is TRUE, and pMonitoredValue is greater than pHighLimit for pTimeDelay, then indicate a transition to the HIGH_LIMIT event state.

(b) If pCurrentState is NORMAL, and the LowLimitEnable flag of pLimitEnable is TRUE, and pMonitoredValue is less than pLowLimit for pTimeDelay, then indicate a transition to the LOW_LIMIT event state.

(c) If pCurrentState is HIGH_LIMIT, and the HighLimitEnable flag of pLimitEnable is FALSE, then indicate a transition to the NORMAL event state.

(d) Optional: If pCurrentState is HIGH_LIMIT, and the LowLimitEnable flag of pLimitEnable is TRUE, and pMonitoredValue is less than pLowLimit for pTimeDelay, then indicate a transition to the LOW_LIMIT event state.

(e) If pCurrentState is HIGH_LIMIT, and pMonitoredValue is less than (pHighLimit - pDeadband) for pTimeDelayNormal, then indicate a transition to the NORMAL event state.

~~(f)~~

(f)~~(g)~~If pCurrentState is LOW_LIMIT, and the LowLimitEnable flag of pLimitEnable is FALSE, then indicate a transition to the NORMAL event state.

(g)~~(h)~~Optional: If pCurrentState is LOW_LIMIT, and the HighLimitEnable flag of pLimitEnable is TRUE, and pMonitoredValue is greater than pHighLimit for pTimeDelay, then indicate a transition to the HIGH_LIMIT event state.

(h)~~(i)~~If pCurrentState is LOW_LIMIT, and pMonitoredValue is greater than (pLowLimit + pDeadband) for pTimeDelayNormal, then indicate a transition to the NORMAL event state.

...

18) The max-segments-accepted parameter of the BACnet-Confirmed-Request-PDU is incorrectly documented as optional (there is no way to not include the parameter so it cannot be optional).

[Change **Clause 20.1.2.4**, p. 615]

20.1.2.4 max-segments-accepted

This ~~optional~~ parameter specifies the maximum number of segments that the device will accept. This parameter is included in the confirmed request so that the responding device may determine how to convey its response. The parameter shall be encoded as follows:

B'000'	Unspecified number of segments accepted.
B'001'	2 segments accepted.
B'010'	4 segments accepted.
B'011'	8 segments accepted.
B'100'	16 segments accepted.
B'101'	32 segments accepted.
B'110'	64 segments accepted.
B'111'	Greater than 64 segments accepted.

19) The Initialize-Routing-Table service is incorrectly referred to as Initialize-Router-Table.

[Change **Clause 6.6.3.9**, p.]

6.6.3.9 Initialize-Routing-Table-Ack

This message is sent by a router after the reception and servicing of an ~~Initialize-Router-Table~~ ***Initialize-Routing-Table*** message. If the router is acknowledging a table update message, signified by a non-zero value in the Number of Ports field, it shall return an Initialize-Routing-Table-Ack without data. If the router is acknowledging a table query message, indicated by a zero value in the Number of Ports field, it shall return a complete copy of its routing table. If a complete copy of the table cannot be returned in a single acknowledgment, the router shall send multiple acknowledgments, each containing a portion of the routing table until the entire table has been sent.

20) In Table K-10, the Feedback_Value property of the Multi-state Output is incorrectly adorned with footnote 1 instead of footnote 2.

[Change **Table K-10**, p. 887]

Table K-10. Properties That AE-AVM-A Devices Shall Be Capable of Presenting and Modifying

All Object Types[1] (from Table K-9)	**Accumulator**	**Analog Objects**
Acked_Transitions[2] Event_State[2] Event_Enable Notification_Class Event_Time_Stamps[2] Time_Delay	Pulse_Rate High_Limit Low_Limit Limit_Monitoring_Interval	Limit_Enable High_Limit Low_Limit Deadband
Binary Input, Binary Value	**Binary Output**	**Event Enrollment**
Alarm_Value	Feedback_Value[2]	Object_Property_Reference Event_Parameters Notify_Type
Loop	**Multi-state Input, Multi-state Value**	**Multi-state Output**
Error_Limit	Alarm_Values Fault_Values	Feedback_Value[~~1~~2]
Notification Class	**Pulse Converter**	
Priority Ack_Required Recipient_List	Limit_Enable High_Limit Low_Limit Deadband	

[1] For object types that include these properties.

[2] AE-AVM-A devices need only be capable of presenting these properties; not modifying them.

21) Table K-11 is incorrectly referred to as Table K11.

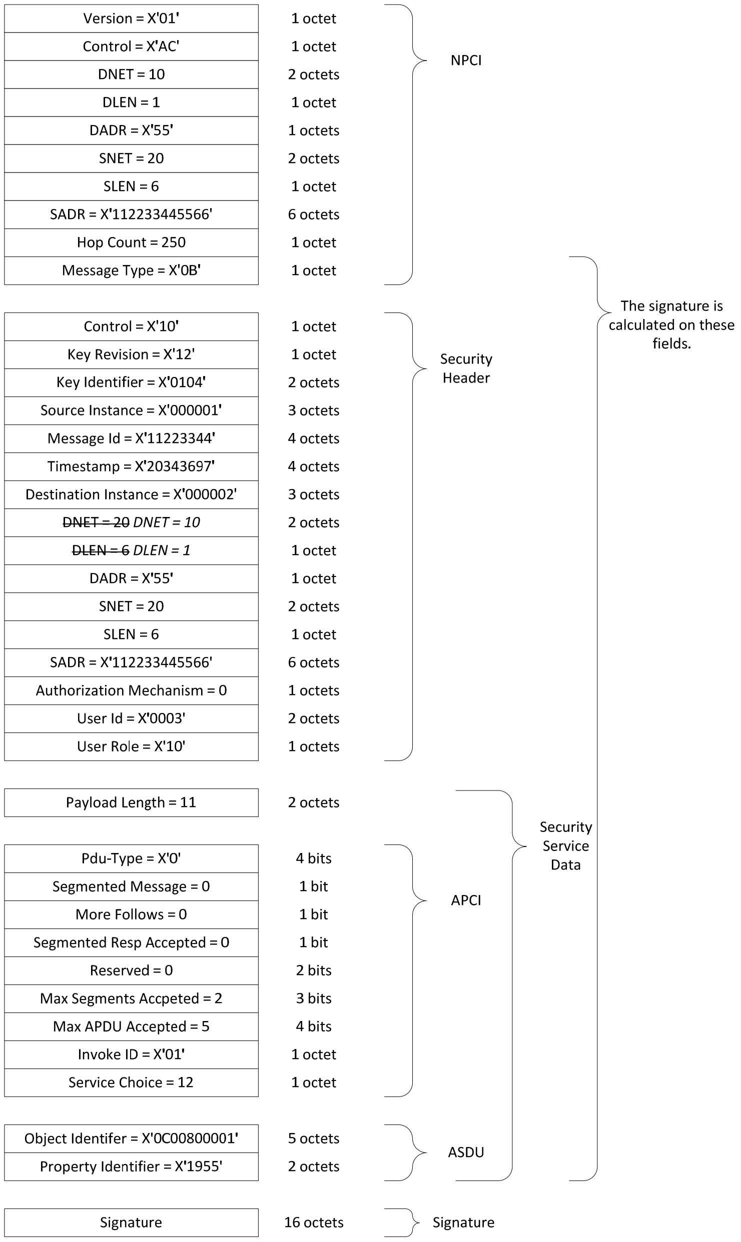
[Change **Table K-11**, p. 889]

Table K-11. Event Log Object Properties That AE-ELVM-A Devices Shall Be Capable of Presenting and Modifying

Enable	Notification_Threshold
Start_Time	Last_Notify_Record (retrieve only)
Stop_Time	Event_State (retrieve only)
Stop_When_Full	Notification_Class
Buffer_Size	Event_Enable
Record_Count	Event_Time_Stamps (retrieve only)
Total_Record_Count (retrieve only)	

22) Figure 24-1 shows an example with an erroneous DNET.

[Change **Figure 24-1**, p. 746]



Version = X'01'	1 octet
Control = X'AC'	1 octet
DNET = 10	2 octets
DLEN = 1	1 octet
DADR = X'55'	1 octets
SNET = 20	2 octets
SLEN = 6	1 octet
SADR = X'112233445566'	6 octets
Hop Count = 250	1 octet
Message Type = X'0B'	1 octet
Control = X'10'	1 octet
Key Revision = X'12'	1 octet
Key Identifier = X'0104'	2 octets
Source Instance = X'000001'	3 octets
Message Id = X'11223344'	4 octets
Timestamp = X'20343697'	4 octets
Destination Instance = X'000002'	3 octets
~~DNET = 20~~ *DNET = 10*	2 octets
~~DLEN = 6~~ *DLEN = 1*	1 octet
DADR = X'55'	1 octet
SNET = 20	2 octets
SLEN = 6	1 octet
SADR = X'112233445566'	6 octets
Authorization Mechanism = 0	1 octets
User Id = X'0003'	2 octets
User Role = X'10'	1 octets
Payload Length = 11	2 octets
Pdu-Type = X'0'	4 bits
Segmented Message = 0	1 bit
More Follows = 0	1 bit
Segmented Resp Accepted = 0	1 bit
Reserved = 0	2 bits
Max Segments Accpeted = 2	3 bits
Max APDU Accepted = 5	4 bits
Invoke ID = X'01'	1 octet
Service Choice = 12	1 octet
Object Identifer = X'0C00800001'	5 octets
Property Identifier = X'1955'	2 octets
Signature	16 octets

Figure 24-1. An example secured APDU (Read-Property).

23) State DONE_WITH_TOKEN in Clause 9.6.5.6 incorrectly sets TokenCount to zero and then to one. As per 135-2004 erratum #2.

[Change **SoleMasterRestartMaintenancePFM** transition in **Clause 9.6.5.6**, p. 106]

SoleMasterRestartMaintenancePFM

If FrameCount is greater than or equal to $\mathbf{N}_{max\_info\_frames}$, and TokenCount is greater than or equal to $\mathbf{N}_{poll}$-1, and (PS+1) modulo ($\mathbf{N}_{max\_master}$+1) is equal to NS, and SoleMaster is TRUE, then set PS to (NS +1) modulo ($\mathbf{N}_{max\_master}$+1) ; call SendFrame to transmit a Poll For Master to PS; set NS to TS (no known successor node); set RetryCount ~~and TokenCount~~ to zero; set TokenCount to one; and enter the POLL_FOR_MASTER state to find a new successor to TS.

24) CONF_SERV is misspelled as CONF_SERVE in Figures 5-8, 5-10, and 5-11.

[Change **Figure 5-8**, p. 46]

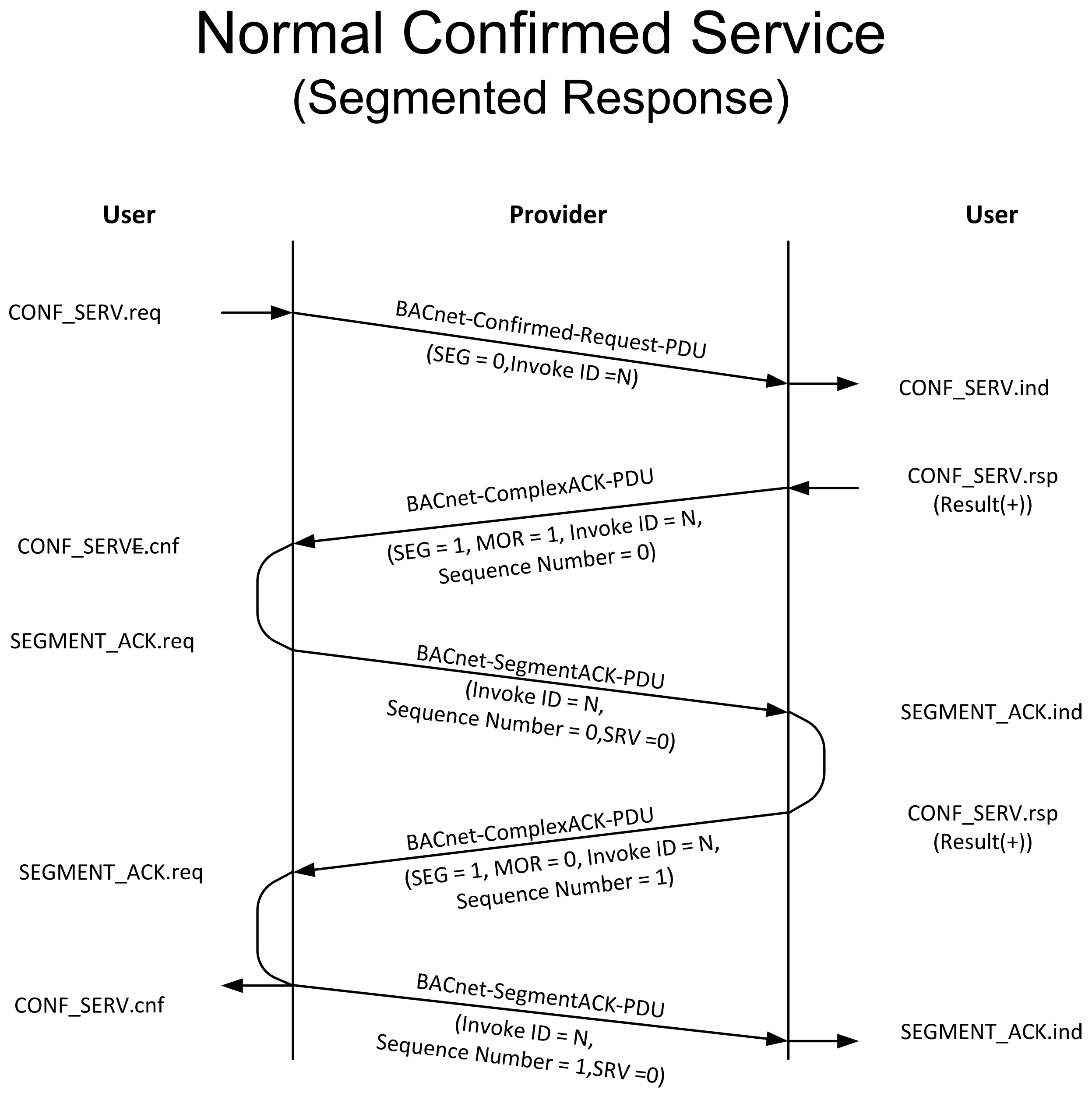


Figure 5-8. Time sequence diagram for a normal confirmed service with segmented response.

[Change **Figure 5-10**, p. 48]

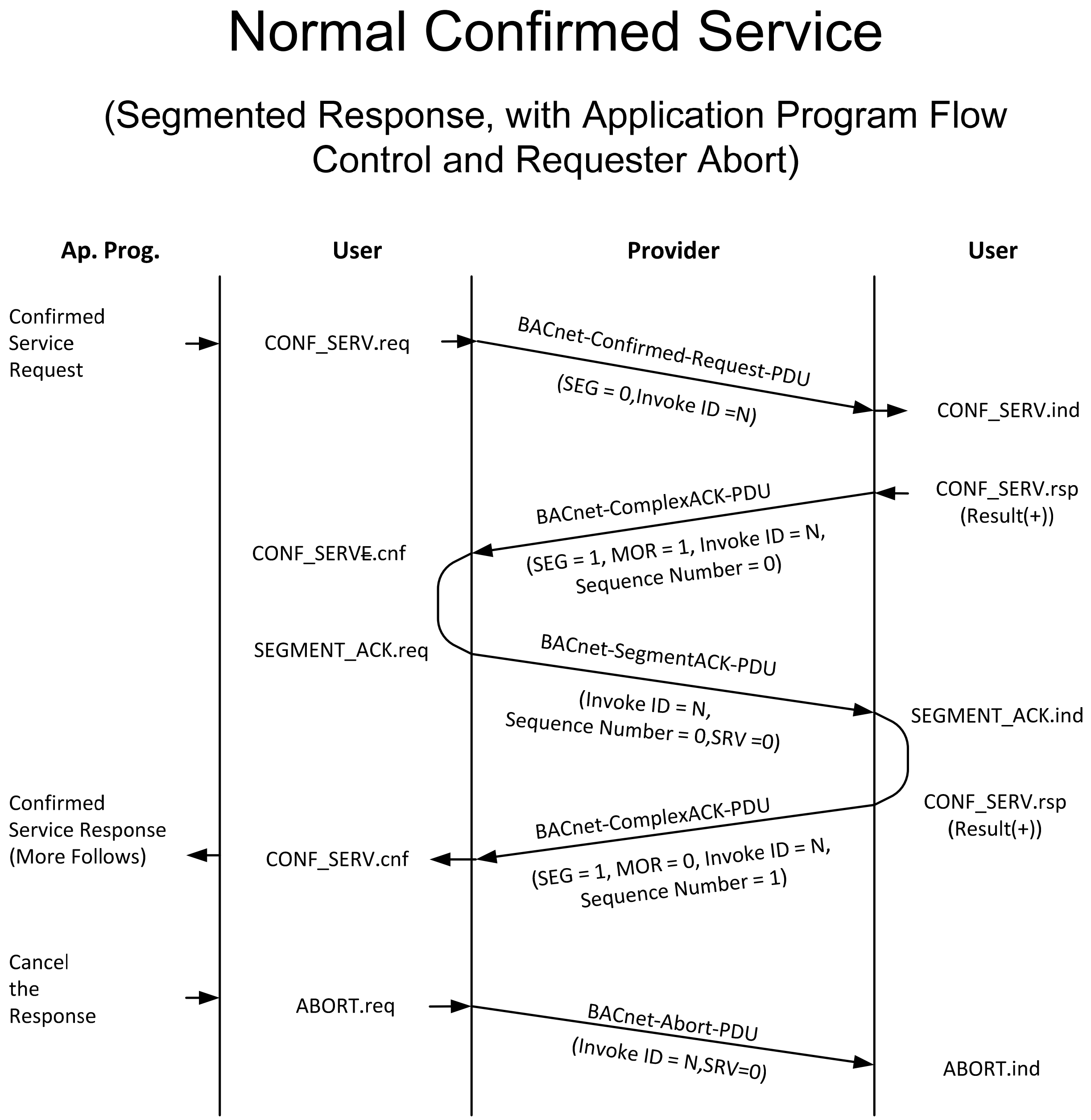

Normal Confirmed Service
(Segmented Response, with Application Program Flow Control and Requester Abort)
Ap. Prog.
User
Provider
User
Confirmed Service Request
CONF_SERV.req
BACnet-Confirmed-Request-PDU
(SEG = 0,Invoke ID =N)
CONF_SERV.ind
CONF_SERV.rsp (Result(+))
BACnet-ComplexACK-PDU
CONF_SERV~~E~~.cnf
(SEG = 1, MOR = 1, Invoke ID = N, Sequence Number = 0)
SEGMENT_ACK.req
BACnet-SegmentACK-PDU
(Invoke ID = N, Sequence Number = 0,SRV =0)
SEGMENT_ACK.ind
CONF_SERV.rsp (Result(+))
Confirmed Service Response (More Follows)
CONF_SERV.cnf
BACnet-ComplexACK-PDU
(SEG = 1, MOR = 0, Invoke ID = N, Sequence Number = 1)
Cancel the Response
ABORT.req
BACnet-Abort-PDU
(Invoke ID = N,SRV=0)
ABORT.ind

Figure 5-10. Time sequence diagram for a normal confirmed service with segmented response, application program flow control, and response cancellation.

[Change **Figure 5-11**, p. 49]

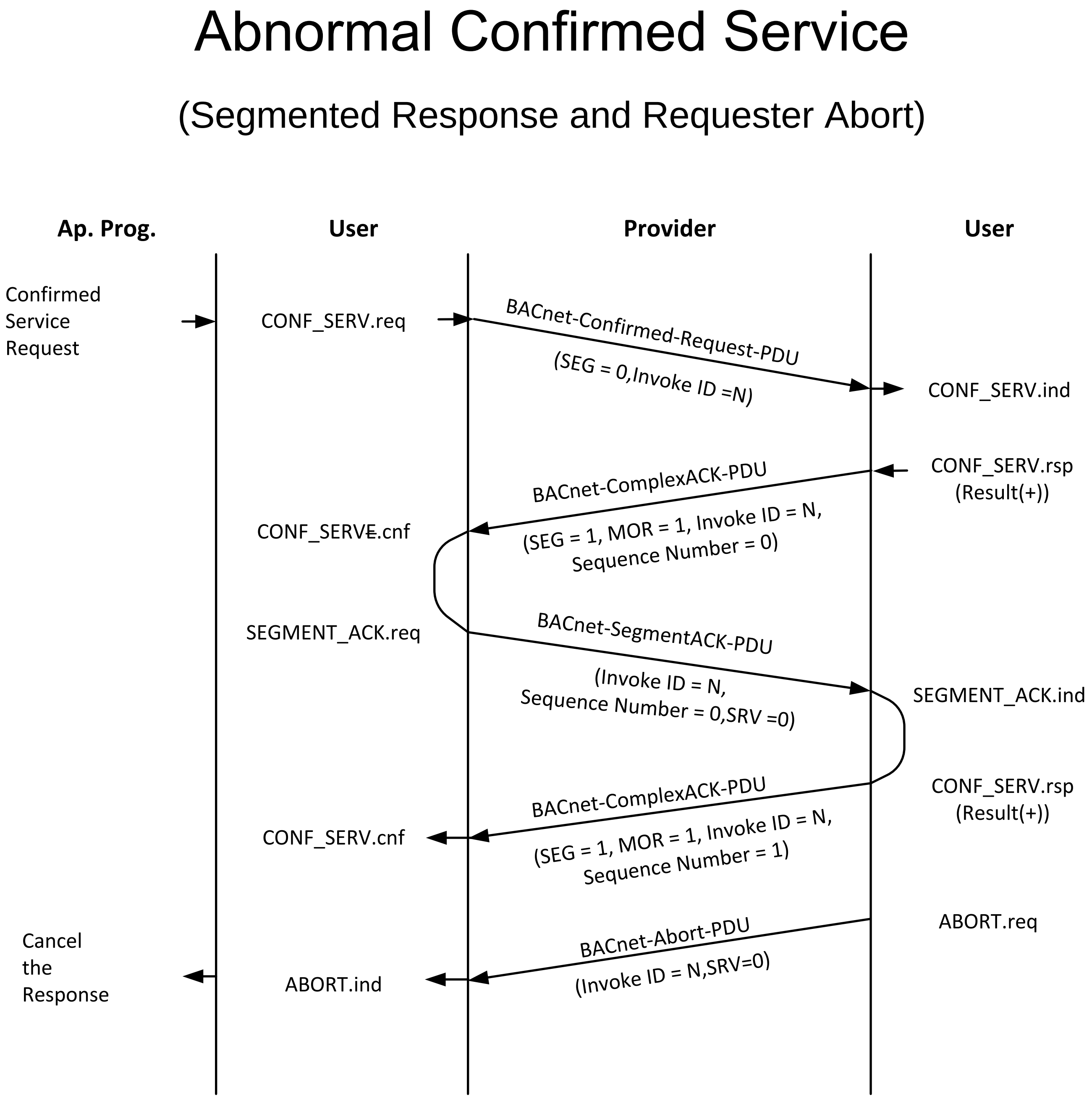

Abnormal Confirmed Service
(Segmented Response and Requester Abort)
Ap. Prog.
User
Provider
User
Confirmed Service Request
CONF_SERV.req
BACnet-Confirmed-Request-PDU
(SEG = 0,Invoke ID =N)
CONF_SERV.ind
CONF_SERV.rsp (Result(+))
BACnet-ComplexACK-PDU
(SEG = 1, MOR = 1, Invoke ID = N, Sequence Number = 0)
CONF_SERV~~E~~.cnf
SEGMENT_ACK.req
BACnet-SegmentACK-PDU
(Invoke ID = N, Sequence Number = 0,SRV =0)
SEGMENT_ACK.ind
CONF_SERV.rsp (Result(+))
BACnet-ComplexACK-PDU
(SEG = 1, MOR = 1, Invoke ID = N, Sequence Number = 1)
CONF_SERV.cnf
ABORT.req
BACnet-Abort-PDU
(Invoke ID = N,SRV=0)
Cancel the Response
ABORT.ind

Figure 5-11. Time sequence diagram for an abnormal confirmed service.

25) The Schedule, Lighting Output, and Channel objects were missed when indicating which properties are reported in CHANGE_OF_RELIABILITY notifications.

[Change **Table 13-5**, p. 473]

Table 13-5. Properties Reported in CHANGE_OF_RELIABILITY Notifications

Object Type	Properties
…	…
Analog Input, Analog Output, Analog Value, Binary Input, Binary Value, BitString Value, ***Channel,*** CharacterString Value, ***Global Group,*** Integer Value, Large Analog Value, ***Lighting Output,*** Multi-state Input, Multi-state Value, Positive Integer Value, Pulse Converter	Present_Value
Schedule	***None***
…	…

26) Clauses 12.21, 12.21.6, 13.8.1.1.6, 13.9.1.1.6, and Annex M incorrectly reference Table 13-5 for event notification priority ranges; instead they should reference Table 13-6.

[Change paragraph 3 of **Clause 12.21**, p. 257]

The purpose of prioritization is to provide a means to ensure that alarms or event notifications with critical time considerations are not unnecessarily delayed. The possible range of priorities is 0 - 255. A lower number indicates a higher priority. The priority and the Network Priority (Clause 6.2.2) are associated as defined in Table *13-6* ~~13-5~~. Priorities may be assigned to TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events individually within a notification class.

[Change **Clause 12.21.6**, p. 258]

12.21.6 Priority
This property, of type BACnetARRAY[3] of Unsigned, shall convey the priority to be used for event notifications for TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively. Priorities shall range from 0 - 255 inclusive. A lower number indicates a higher priority. The priority and the Network Priority (see 6.2.2) are associated as defined in Table *13-6* ~~13-5~~.

[Change **Clause 13.8.1.1.6**, p. 515]

13.8.1.1.6 Priority
This parameter, of type Unsigned8, shall specify the priority of the event that has occurred. The priority is specified by the Priority property of the Notification Class or Event Enrollment objects associated with this event. The possible range of priorities is 0-255. A lower number indicates a higher priority. The priority and the Network Priority (see 6.2.2) are associated as defined in Table *13-6* ~~13-5~~.

[Change **Clause 13.8.1.1.6**, p. 515]

13.9.1.1.6 Priority

This parameter, of type Unsigned8, shall specify the priority of the event that has occurred. The priority is specified by the Priority property of the Notification Class object associated with the event. The possible range of priorities is 0-255. A lower number indicates a higher priority. The priority and the Network Priority (see 6.2.2) are associated as defined in Table *13-6* ~~13-5~~.

[Change **ANNEX M**, p. 911]

ANNEX M - GUIDE TO EVENT NOTIFICATION PRIORITY ASSIGNMENTS (INFORMATIVE)

[This annex is not part of this standard. It is merely informative and does not contain requirements for conformance to the standard.]

The Alarm and Event Priorities and Network Priorities defined in 13.4.1 broadly categorize the alarm and event notification priorities. This annex provides examples of various alarms and events that could be assigned into these categories.

Table M-1 extends Table *13-6* ~~13-5~~ by adding semantic meaning to the priority classifications. The subsequent narrative details the classifications and provides examples of various alarm and event priorities in an interoperable system.

...

27) Clause 13.8.1.1.6 still refers to the Priority property of the Event Enrollment object which was removed in Protocol Revision 4.

[Change **Clause 13.8.1.1.6**, p. 515]

13.8.1.1.6 Priority

This parameter, of type Unsigned8, shall specify the priority of the event that has occurred. The priority is specified by the Priority property of the Notification Class ~~or Event Enrollment objects~~ *object* associated with this event. The possible range of priorities is 0-255. A lower number indicates a higher priority. The priority and the Network Priority (see 6.2.2) are associated as defined in Table *13-6* ~~13-5~~.

28) Clause S.1 incorrectly documents the fields of a Request-Master-Key message.

[Change the first entry in the first table in **Clause S.1**, p. 1012]

; Send a request for a Device-Master key. Request-Master-Key(Control = NPDU, Key Revision = 0, Key Id = 0/0, Source Device Instance = SecDev1, Message Id = any valid value, Timestamp = any valid value (may be incorrect), Destination Device Instance = 4194303, DNET = 65535, DADR = empty, SNET = 0, SADR = SecDev1MAC, Authentication Mechanism = not present, Authentication Data = not present, Service Data = ~~Number_Of_Encryption_Algs = 1,~~ ~~Supported_Encryption_Algs = (0),~~	

~~Number_Of_Signature_Algs = 2,~~ ~~Supported_Signature_Algs = (0, 1),~~ ***Number of Supported Key Algorithms = 2,*** ***Encryption and Signature Algorithms = (0, 1),*** Padding = not present, Signature = all 0s)	

29) In the description of RS-485 network medium there is a grammatical error.

[Change **Clause 9.2.1**, p. 81]

9.2.1 Medium

An MS/TP EIA-485 network shall use shielded, twisted-pair cable for data signaling with characteristic impedance between 100 and 130 ohms. Distributed capacitance between conductors shall be less than 100 pF per meter (30 pF per foot). Distributed capacitance between conductors and shield shall be less ~~that~~ *than* 200 pF per meter (60 pF per foot). Foil or braided shields are acceptable. The maximum recommended length of an MS/TP segment with AWG 18 (0.82 mm2 conductor area) cable is specified in Clause 9.2.3. The use of greater distances and/or different wire gauges shall comply with the electrical specifications of EIA-485.

30) A number of clauses incorrectly reference Table 13-5 where they should be referencing Table 13-6.

[Change 3rd paragraph of **Clause 12.21,** p. 257]

The purpose of prioritization is to provide a means to ensure that alarms or event notifications with critical time considerations are not unnecessarily delayed. The possible range of priorities is 0 - 255. A lower number indicates a higher priority. The priority and the Network Priority (Clause 6.2.2) are associated as defined in Table 13-~~5~~*6*. Priorities may be assigned to TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events individually within a notification class.

[Change **Clause 12.21.6**, p. 258]

12.21.6 Priority

This property, of type BACnetARRAY[3] of Unsigned, shall convey the priority to be used for event notifications for TO_OFFNORMAL, TO_FAULT, and TO_NORMAL events, respectively. Priorities shall range from 0 - 255 inclusive. A lower number indicates a higher priority. The priority and the Network Priority (see 6.2.2) are associated as defined in Table 13-~~5~~*6*.

[Change **Clause 13.8.1.1.6**, p. 515]

13.8.1.1.6 Priority

This parameter, of type Unsigned8, shall specify the priority of the event that has occurred. The priority is specified by the Priority property of the Notification Class or Event Enrollment objects associated with this event. The possible range of priorities is 0-255. A lower number indicates a higher priority. The priority and the Network Priority (see 6.2.2) are associated as defined in Table 13-~~5~~*6*.

[Change **Clause 13.9.1.1.6**, p. 518]

13.9.1.1.6 Priority

This parameter, of type Unsigned8, shall specify the priority of the event that has occurred. The priority is specified by the Priority property of the Notification Class object associated with the event. The possible range of priorities is 0-255. A lower number indicates a higher priority. The priority and the Network Priority (see 6.2.2) are associated as defined in Table 13-~~5~~*6*.

[Change **ANNEX M**, p. 911]
[This change also includes a change to the referenced clause containing Table 13-6]

ANNEX M - GUIDE TO EVENT NOTIFICATION PRIORITY ASSIGNMENTS (INFORMATIVE)

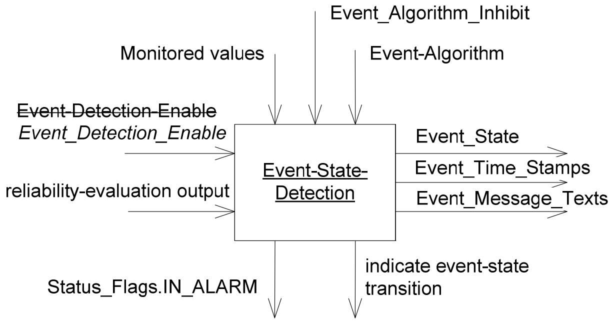
[This annex is not part of this standard. It is merely informative and does not contain requirements for conformance to the standard.]

The Alarm and Event Priorities and Network Priorities defined in *13.2.5.1*~~13.4.1~~ broadly categorize the alarm and event notification priorities. This annex provides examples of various alarms and events that could be assigned into these categories.

Table M-1 extends Table 13-~~5~~*6* by adding semantic meaning to the priority classifications. The subsequent narrative details the classifications and provides examples of various alarm and event priorities in an interoperable system.

31) Figure 13.2.2 incorrectly references Event-Detection-Enable instead of Event_Detection_Enable.

[Change **Figure 13.2.2**, p. 466]



32) Clause 13.2.2.1.5 contains a superfluous *the*.

[Change the 2nd paragraph of **Clause 13.2.2.1.5**, p. 469]

Upon Event_Algorithm_Inhibit changing to TRUE, the event shall transition to the NORMAL state if not already there. While Event_Algorithm_Inhibit remains TRUE, no transitions shall occur except those into and out of FAULT. Upon Event_Algorithm_Inhibit changing to FALSE, any condition shall hold for ~~the~~ its regular time delay after the change to FALSE before a transition is generated.

33) The BACnetAccessEvent production in clause 21 incorrectly names *no-entry-after-granted* as *no-entry-after-grant*.

[Change **Clause 21**, p. 662]

BACnetAccessEvent ::= ENUMERATED {
…
~~no-entry-after-grant~~ *no-entry-after-granted* (16),
...

34) The description of Out_Of_Service for the Global Group object has the description backwards.

[Change **Clause 12.20.12**, p. 425]

12.50.12 Out_of_Service

This property, of type BOOLEAN, indicates and controls whether (TRUE) or not (FALSE) the Present_Value property is *decoupled and is not* updated to track the values of the group members. In addition, the Reliability property and the corresponding state of the FAULT flag of the Status_Flags property shall be decoupled from their normal calculations when Out_Of_Service is TRUE. While the Out_Of_Service property is TRUE, the Reliability property may be changed to any value as a means of simulating specific fixed conditions or for testing purposes. Other functions that depend on the state of the Reliability property shall respond to changes made to these properties while Out_Of_Service is TRUE as if those changes had occurred by normal operation.

35) Some XML closing tags are incorrectly documented as opening tags.

[Change **Clause Q.3.4.1**, p. 981]

...

```
 <Definitions>
       <BitString name="999-WidgetStatusFlags" length="2">
             <NamedBits>
                   <Bit bit="0" name="too-hot"/>
                   <Bit bit="1" name="too-cold"/>
             <NamedBits>
             <NamedValues>
                   <BitString name="ok" displayName="All is well" value="">
                   <BitString name="error" displayName="Confused" value="too-hot;too-cold">
             ~~<NamedValues>~~
             *</NamedValues>*
       </BitString>
 </Definitions>
```

...

[Change **Clause Q.3.4.2**, p. 982]

...

```
<Definitions>
       <BitString name="0-BACnetLogStatus" length="3">
             <NamedBits>
                   <Bit bit="0" name="log-disabled" displayName="Disabled"/>
                   <Bit bit="1" name="buffer-purged" displayName="Purged"/>
                   <Bit bit="2" name="log-interrupted" diaplayName="Interrupted"/>
             ~~<NamedBits>~~
             *</NamedBits>*
       </BitString>
</Definitions>
```

36) The standard range for the BACnetDoorAlarmState is inconsistent between Clause 23 and Clause 21.

[Change **BACnetDoorAlarmState production** in **Clause 21**, p. 669]

```
 BACnetDoorAlarmState ::= ENUMERATED {
       normal              (0),
       alarm               (1),
       door-open-too-long    (2),
       forced-open         (3),
       tamper              (4),
       door-fault          (5),
       lock-down           (6),
       free-access         (7),
       egress-open         (8),
       ...
```

}
-- Enumerated values 0-~~63~~*255* are reserved for definition by ASHRAE. Enumerated values
-- ~~64~~*256*-65535 may be used by others subject to the procedures and constraints described
-- in Clause 23

37) The BACnetLightingOperation and BACnetLightingTransition productions as missing the ellipsis indicating that the productions are extensible.

[Change **BACnetLightingOperation production** in **Clause 21**, p. 685]

```
BACnetLightingOperation ::= ENUMERATED {
     none            (0),
     fade-to         (1),
     ramp-to         (2),
     step-up         (3),
     step-down       (4),
     step-on         (5),
     step-off        (6),
     warn            (7),
     warn-off        (8),
     warn-relinquish   (9),
     stop            (10),
     ...
     }
     -- Enumerated values 0-255 are reserved for definition by ASHRAE. Enumerated values 256-65535 may be used by
     -- others subject to the procedures and constraints described in Clause 23.
```

[Change **BACnetLightingTransition production** in **Clause 21**, p. 685]

```
BACnetLightingTransition ::= ENUMERATED {
     none     (0),
     fade     (1),
     ramp     (2),
     ...
     }
     -- Enumerated values 0-63 are reserved for definition by ASHRAE. Enumerated values 64-255 may be used by
     -- others subject to the procedures and constraints described in Clause 23.
```

38) The ReadRange service description incorrectly implies that the service works on to arrays which are not arrays of lists.

[Change **Clause 15.8.1.2.4**, p. 556]

15.8.1.2.4 Result Flags

This parameter, of type BACnetResultFlags, shall convey several flags that describe characteristics of the response data:

{FIRST_ITEM, LAST_ITEM, MORE_ITEMS}

The FIRST_ITEM flag indicates whether this response includes the first list ~~or array~~ element (in the case of positional indexing), or the oldest timestamped item (in the case of time indexing).

The LAST_ITEM flag indicates whether this response includes the last list ~~or array~~ element (in the case of positional indexing), or the newest timestamped item (in the case of time indexing)

The MORE_ITEMS flag indicates whether more items matched the request but were not transmittable within the PDU.

[Change **Clause 15.1.8.1.2.4**, p. 556]

15.8.2 Service Procedure

The responding BACnet-user shall first verify the validity of the 'Object Identifier', 'Property Identifier' and 'Property Array Index' parameters and return a 'Result(-)' response with the appropriate error class and code if the object or property is unknown, if the referenced data is not a list or array *of lists*, or if it is currently inaccessible for another reason.

If the 'Range' parameter is not present, then the responding BACnet-user shall read and attempt to return all of the available items in the list~~ or array~~.

...

39) Clauses K.5.7 and K.5.8 were removed but were not marked as deleted as clauses K.1.5 and K.1.6 were.

[Insert **Clauses K.5.7** and **K.5.8**, p. 898]

K.5.7 Deleted Clause

This clause has been removed.

K.5.8 Deleted Clause

This clause has been removed.

40) The BUFFER_READY algorithm description should restrict the threshold parameter to Unsigned32.

[Change **Clause 13.8.2**, p. 562]

13.3.7 BUFFER_READY Event Algorithm

…

pThreshold This parameter, of type ~~Unsigned~~*Unsigned32*, represents the number of records that, when added to the log buffer, will result in a transition to NORMAL. If this parameter has a value of 0, then no transitions will be indicated by the algorithm.

…

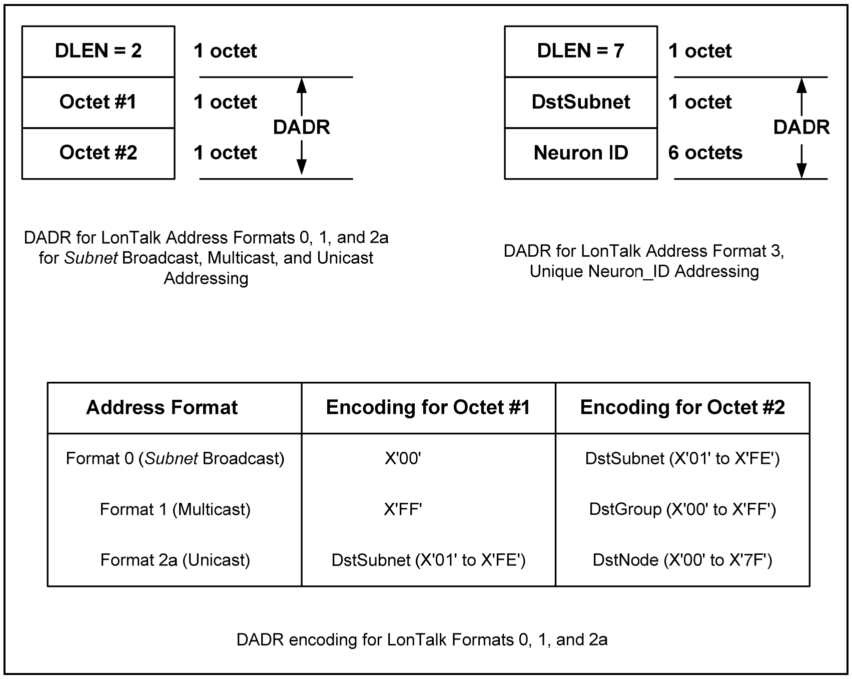
41) The descriptions of LonTalk broadcasts are misleading in **Table 6-2** and **Figure 6-4**.

[Change **Table 6-2**, p. 58]

Table 6-2. BACnet DADR and SADR Encoding Rules Based Upon Data Link Layer Technology

BACnet Data Link Layer	DLEN	SLEN	Encoding Rules
ISO 8802-3 ("Ethernet"), as defined in Clause 7	6	6	Encoded as in their MAC layer representations
ARCNET, as defined in Clause 8	1	1	Encoded as in their MAC layer representations
MS/TP, as defined in Clause 9	1	1	Encoded as in their MAC layer representations
LonTalk ~~domain wide~~ ***subnet*** broadcast	2	2	The encoding for the SADR is shown in Figure 6-3 The encoding for the DADR is shown in Figure 6-4
LonTalk multicast	2	2	
LonTalk unicast	2	2	
LonTalk, unique Neuron_ID	7	2	
BACnet/IP, as defined in Annex J	6	6	Encoded as specified in J.1.2
ZigBee, as defined in Annex O	3	3	A VMAC Address encoded as a device instance as shown in Annex H.7 Virtual MAC Addressing

[Change **Figure 6-4**, p. 58]


DLEN = 2
1 octet
Octet #1
1 octet
Octet #2
1 octet
DADR
DLEN = 7
1 octet
DstSubnet
1 octet
Neuron ID
6 octets
DADR
DADR for LonTalk Address Formats 0, 1, and 2a for *Subnet* Broadcast, Multicast, and Unicast Addressing
DADR for LonTalk Address Format 3, Unique Neuron_ID Addressing
DADR encoding for LonTalk Formats 0, 1, and 2a

Address Format	Encoding for Octet #1	Encoding for Octet #2
Format 0 (*Subnet* Broadcast)	X'00'	DstSubnet (X'01' to X'FE')
Format 1 (Multicast)	X'FF'	DstGroup (X'00' to X'FF')
Format 2a (Unicast)	DstSubnet (X'01' to X'FE')	DstNode (X'00' to X'7F')

Figure 6-4. Encoding of the DLEN and DADR for NPDUs destined for LonTalk devices being routed through BACnet. The different LonTalk address formats are encoded as shown.

42) In Clause 5.4.5.1 in the ConfirmedSegmentedReceived transition of the IDLE state, the action incorrectly identifies the packet received as a BACnet-ComplexACK-PDU.

[Change **Clause 5.4.5.1**, p. 36]

ConfirmedSegmentedReceived

If a BACnet-Confirmed-Request-PDU whose 'segmented-message' parameter is TRUE, whose 'sequence-number' parameter is zero, and whose 'proposed-window-size' is greater than zero and less than or equal to 127 is received from the network layer and the local device supports the reception of segmented messages,

then save the ~~BACnet-ComplexACK-PDU~~ *BACnet-Confirmed-Request-PDU* segment; compute ActualWindowSize based on the 'proposed-window-size' parameter of the received BACnet-Confirmed-Request-PDU and on local conditions; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-SegmentACK-PDU with 'negative-ACK' = FALSE, 'server' = TRUE, and 'actual-window-size' = ActualWindowSize; start SegmentTimer; set LastSequenceNumber to zero; set InitialSequenceNumber to zero; set DuplicateCount to zero; and enter the SEGMENTED_REQUEST state to receive the remaining segments. (The method used to determine ActualWindowSize is a local matter, except that the value shall be less than or equal to the 'proposed-window-size' parameter of the received BACnet-Confirmed-Request-PDU and shall be in the range 1 to 127, inclusive.)

43) In Clause 12.54, footnote 2 for Table 12.64 has a typo.

[2] This property is required if, and shall be present ~~on~~ *only* if, the object supports COV reporting.