BACnet *Errata*
ANSI/ASHRAE STANDARD 135-2016
A Data Communication Protocol for Building Automation and Control Networks

July 6, 2020

This document lists all known *errata* to ANSI/ASHRAE Standard 135-2016 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 back page marking identifying the electronic publication of Standard 135-2016 is “Product code: D-86451 6/16”.

Changes are indicated by using ~~strikeout~~ for text to be removed and *italics* for text to be added, unless noted otherwise. Grey highlighting is used for marking small corrections.

1) **Clause 4.1**, p. 12: The list of standard datalink layer options for BACnet is misformatted.

	Ethernet (ISO 8802-3)	Clause 7		
	ARCNET (ATA 878.1)	Clause 8		
	MS/TP	Clause 9		
	PTP	Clause 10		
	LonTalk (ISO/IEC 14908.1)	Clause 11		
	BACnet/IP	Annex J		
	BACnet/IPv6	Annex U	~~ZigBee~~	~~Annex~~
~~O~~	*ZigBee*	*Annex O*		

2) **Table 13-7**, p. 602: UNSIGNED RANGE is misplaced outside the table. The reference to Clause 13.3.9 is not shown.

Table 13-7. Standardized Event Algorithms

Event Algorithm	Clause
NONE	13.3.17
...	...
UNSIGNED_OUT_OF_RANGE	13.3.15
UNSIGNED_RANGE	*13.3.9*

~~UNSIGNED_RANGE~~

3) **Clause K.2.16, 2nd**, p. 1055: Repeated word "the".

K.2.16 BIBB - Alarm and Event Management-View and Modify-A (AE-VM-A)

Device A displays and modifies limits and related parameters in standard event-initiating objects.

Device A shall support DS-RP-A and DS-WP-A. The A device shall be capable of using ReadProperty to retrieve and WriteProperty to modify any of the ~~the~~ event and fault algorithm parameters listed in Tables K-11 and K-12. Such parameters may be present in individual properties, in event parameter properties, or in fault parameter properties. See the respective property specifications. Device A may use alternate services where support for execution of the alternate service is supported by Device B.

...

4) **Clause K.2.16, 3rd §**, p. 1055: The word "writing" is missing.

...

Devices claiming conformance to AE-VM-A shall be capable of reading, presenting and *writing* all standard properties in standard objects that are configuration parameters of standard event and/or fault algorithms that have high and low numerical limits, as listed in Tables K-11 and K-12.

5) **Clause K.1.25 and K.1.26**, p. 1055: The word "Multiple" is misspelled in the title.

K.1.25 BIBB - Data Sharing-Change Of Value ~~Mulitple~~ *Multiple*-A (DS-COVM-A)

K.1.26 BIBB - Data Sharing-Change Of Value ~~Mulitple~~ *Multiple*-B (DS-COVM-B)

6) **Clause L.1, 1st §**, p. 1079: The second sentence is unclear.

L.1 Operator Interface Profiles

The following table indicates which BIBBs shall be supported by the device types of this family, for each interoperability area. ~~The B-XAWS is excluded from this table.~~ *The B-XAWS is not shown in this table and is described in Clause L.1.1.*

...

7) **Table 12-71**, p. 518,
Clause 12.56.32, p. 533,
Clause J.4.3.2, p. 1028, and
Clause J.4.5, 2nd §, p. 1029: The references to J.7.8 are incorrect.

Table 12-71. Properties of the Network Port Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
...	...	...
Profile_Name	CharacterString	O

[1] ...

...

[9] Required to be present if the Network_Type is IPV4, Protocol_Level is BACNET_APPLICATION, and the device is capable of communicating through a NAT router as described in Clause ~~J.7.8~~ *J.7.5.*

[10] Required if Network_Type is IPV4, Protocol_Level is BACNET_APPLICATION, and the device is configured to communicate through a NAT router as described in Clause ~~J.7.8~~ *J.7.5.*

... ...

12.56.32 BACnet_IP_NAT_Traversal

This property, of type BOOLEAN, indicates whether (TRUE) or not (FALSE) this port is configured to operate in a NAT environment, as described in Clause ~~J.7.8~~ *J.7.5*, and the global address is indicated by the value of the BACnet_IP_Global_Address property.

...

J.4.3.2 Broadcast Distribution Table Format

The BDT consists of one entry for the address of the BBMD for the local IP subnet and an entry for the BBMD on each remote IP subnet to which broadcasts are to be forwarded. Each entry consists of the 6-octet B/IP address with which the BBMD is accessed and a 4-octet broadcast distribution mask. If the IP router to the subnet performs network address translation (NAT), then the BDT entry shall contain the global IP address of the IP router. The operation of BBMDs in the presence of NAT is described in Clause ~~J.7.8~~ *J.7.5*. If messages are to be distributed on the remote IP subnet using directed broadcasts, the broadcast ...

J.4.5 BBMD Operation - Broadcast Distribution

...
address as well as to each foreign device currently in the BBMD's FDT. A BBMD on a subnet with no other BACnet devices (such as a NAT-supporting BBMD, see Clause ~~J.7.8~~ *J.7.5*) may omit the broadcast using the B/IP broadcast address. The method by which a BBMD determines whether or not other BACnet devices are present is a local matter.

8) **Clause W.10, 2nd §**, p. 1209: The word "it" on first line of 2nd paragraph should be "its".

W.10 Representation of Metadata

If metadata items are represented as part of their associated data item, they are represented in a brief "short form" because their type is fixed and therefore assumed. The "short form" encodes only the metadata name and its value or children.

For example, when included along with ~~it~~ *its* associated data item, (e.g., /path/to/example), the 'minimum' metadata is encoded as:

9) **Clause 12.56.55, 2nd §**, p. 537: The reference to Clause 12 is misleading.

12.56.55 Auto_Slave_Discovery

...

Slave detection shall be accomplished by the proxy device using ReadProperty services to read, at a minimum, the Device object's Protocol_Services_Supported property for each MAC address on the network connected to this port. The ReadProperty service shall use the special object instance of 4194303 as described in ~~Clause 12~~ *Clause 15.5.2*. If the device is found to support execution of the Who-Is service, it is ignored; otherwise, the device shall be added to the Slave_Address_Binding property. The slave detection algorithm shall be repeated periodically. The period at which it is repeated is a local matter.

10) **Clause 21, BACnetPropertyIdentifier**, p. 845: The word "see" is missing in the numeric index for the Reliability property.

```
BACnetPropertyIdentifier ::= ENUMERATED {  -- see below for numerical order
          absentee-limit                         (244),
          ...                                    ...
          zone-to                                (321),
-- -numerical order reference
          -- see acked-transitions               (0),
          ...                                    ...
          -- see reliability                     (103),
          ...
...
```

11) **Table W-14**, p. 1241: The error code WS_ERR_UNINITIALIZED is incorrect.

Table W-14. Error Numbers

Error Name	Error Number	HTTP Status Code	Example Error Text
...	...	...	...
WS_ERR_UNINITIALIZED_*VALUE*	43	403	"Data is uninitialized and has no value"
...	...	...	...

12) **Clause 12.10.22**, Command object type, p. 209,
Clause 12.11.68, Device object type, p. 220,
Clause 12.12.25, Event Enrollment object type, p. 228,
Clause 12.21.20, Notification Class object type, p. 288,
Clause 12.25.37, Trend Log object type, p. 307,
Clause 12.27.29, Event Log object type, p. 322,
Clause 12.28.32, Load Control object type, p. 332,
Clause 12.30.35, Trend Log Multiple object type, p. 346,
Clause 12.33.15, Access User object type, p. 375,
Clause 12.34.12, Access Rights object type, p. 380, and
Clause 12.35.26, Access Credential object type, p. 389: The Reliability_Evaluation_Inhibit property language in these clauses refers to the Out_Of_Service property that does not exist in these object types.

12.X.Y Reliability_Evaluation_Inhibit

This property, of type BOOLEAN, indicates whether (TRUE) or not (FALSE) reliability-evaluation is disabled in the object. This property is a runtime override that allows temporary disabling of reliability-evaluation.

When reliability-evaluation is disabled, the Reliability property shall have the value NO_FAULT_DETECTED ~~unless Out_Of_Service is TRUE and an alternate value has been written to the Reliability property~~.

13) **Clause 15.7.3.1.2**, p. 690,
Clause 21, ReadAccessResult production, p. 870: The language on what to return for property identifier OPTIONAL misses the case of no optional properties are present, and no other properties were requested, so that no property value is returned for the object.

15.7.3.1.2 List of Property References

...

The property identifier REQUIRED means that only those standard properties having a conformance code of "R" or "W" shall be returned. The property identifier OPTIONAL means that only those standard properties present in the object that have a conformance code "O" shall be returned. *If no optional properties are present then no entries associated with the OPTIONAL property identifier shall be in the 'List of Results' and if no other properties were requested, then the 'List of Results' shall be present and empty.* The Property_List property shall not be returned when properties ALL or REQUIRED are requested. See the specification for the particular object type in Clause 12. If the property identifier ALL, REQUIRED, or OPTIONAL is specified and any of the selected properties is not readable by this service, then a Property Access Error for that property shall be returned in the List of Read Access Results as specified by Clause 15.7.3.2.

```
ReadAccessResult ::= SEQUENCE {
   object-identifier              [0] BACnetObjectIdentifier,
   list-of-results                [1] SEQUENCE OF SEQUENCE {
            property-identifier           [2] BACnetPropertyIdentifier,
            property-array-index          [3] Unsigned OPTIONAL, -- used only with array datatype
                                                                 -- if omitted with an array the entire
                                                                 -- array is referenced
            read-result           CHOICE  {
                                          property-value           [4] ABSTRACT-SYNTAX.&Type,
                                          property-access-error    [5] Error
                                          }
            } ~~OPTIONAL~~
   }
```

14) **Clause 21, BACnetPropertyIdentifier**, p. 846: "base-device-security-policy (327)," is misplaced.

```
BACnetPropertyIdentifier ::= ENUMERATED {  -- see below for numerical order
   ...
   bacnet-ipv6-udp-port                      (438),
   bacnet-ipv6-multicast-address             (440),    ~~base-device-security-policy                    (327),~~
   base-device-security-policy               (327),
   bbmd-accept-fd-registrations              (413),
   ...
```

15) **Table 12-15**, p. 223: There are missing lines in Table 12-15.

Table 12-15. Event Algorithm, Event Parameters and Event Algorithm Parameters

Event Algorithm	Event Parameters	Event Algorithm Parameters
NONE	none	none
ACCESS_EVENT	List_Of_Access_Events Access_Event_Time_Reference	pAccessEvents Referent's value is pAccessEventTime
...	...	...
OUT_OF_RANGE	Time_Delay Low_Limit High_Limit Deadband	pTimeDelay pLowLimit pHighLimit pDeadband
SIGNED_OUT_OF_RANGE	Time_Delay Low_Limit High_Limit Deadband	pTimeDelay pLowLimit pHighLimit pDeadband
...	...	...

16) **Clauses L.2, L.3 and L.6**, p. 1082, 1084 and 1091: The device profiles B-ALSWS, B-LSWS, B-AACWS, and B-AACC erroneously require NM-CE-A, which was removed from all device profiles with addendum 135-2008*v*.

L.2 Life Safety Operator Interface Profiles

The following table indicates which BIBBs shall be supported by the device types of this family, for each interoperability area.

Device & Network Management

B-ALSWS	**B-LSWS**	**B-LSAP**
DM-DDB-A,B	DM-DDB-A,B	DM-DDB-A,B
DM-ANM-A		
DM-ADM-A		
DM-DOB-B	DM-DOB-B	DM-DOB-B
DM-DCC-A	DM-DCC-A	
DM-MTS-A	DM-MTS-A	
DM-OCD-A		
DM-RD-A	DM-RD-A	
DM-BR-A	DM-BR-A	
~~NM-CE-A~~	~~NM-CE-A~~	

L.2.1 BACnet Advanced Life Safety Workstation (B-ALSWS)

...

Device and Network Management

...

- ~~Ability to command half-routers to establish and terminate connections~~

L.2.2 BACnet Life Safety Workstation (B-LSWS)

...

Device and Network Management

...

- ~~Ability to command half-routers to establish and terminate connections~~

L.3 Access Control Operator Interface Profiles

The following table indicates which BIBBs shall be supported by the device types of this family, for each interoperability area.

Device & Network Management

B-AACWS	**B-ACWS**	**B-ACSD**
DM-DDB-A,B	DM-DDB-A,B	DM-DDB-A,B
DM-ANM-A		
DM-ADM-A		
DM-DOB-B	DM-DOB-B	DM-DOB-B
DM-DCC-A	DM-DCC-A	
DM-MTS-A	DM-MTS-A	
DM-OCD-A	DM-OCD-A	
DM-RD-A		
DM-BR-A		
~~NM-CE-A~~		

...

L.3.1 BACnet Advanced Access Control Workstation (B-AACWS)

...

Device and Network Management

...

~~• Ability to command half-routers to establish and terminate connections~~

L.6 Access Control Controller Profiles

The following table indicates which BIBBs shall be supported by the device types of this family, for each interoperability area.

Device & Network Management

B-AACC	**B-ACC**
DM-DDB-A,B	DM-DDB-A,B
DM-DOB-B	DM-DOB-B
DM-DCC-B	DM-DCC-B
DM-TS-B or DM-UTC-B	DM-TS-B or DM-UTC-B
DM-RD-B	DM-RD-B
DM-BR-B	
~~NM-CE-A~~	

L.6.1 BACnet Advanced Access Control Controller (B-AACC)

...

Device and Network Management
...
~~• Ability to command half-routers to establish and terminate connections~~

17) **Clause 19.5.1.3, 5th §**, p. 756: ReadProperty should be WriteProperty.

After commanding or writing the Present_Value of an object, the device, and only the device which wrote or commanded the value, may update the value source information, at the same priority as the command in case of commanding, to set the source device instance or to indicate the object that initiated the operation. The writing or commanding device accomplishes this by writing to the Value_Source property at the same priority that the Present_Value was written or commanded at. Writing to the Value_Source property may be requested by a subsequent ~~ReadProperty~~ *WriteProperty* request or, when WritePropertyMultiple is used, by a respective property value subsequent to the property value written or commanded. Attempts to write to the Value_Source property by a device other than the device that wrote the property or commanded the property at a particular priority shall cause an error being returned and the write to Value_Source shall not be performed.

18) **Table W-10**, p. 1233: This table fails to show the "stagger" parameter defined in last paragraph of W.36.1.

Table W-10. ".subs" Data Items

Path	Type	Description
{prefix}/.subs	Collection	The list of active subscriptions
. . .	. . .	. . .
{prefix}/.subs/{id}/logs/{n}/frequency	Enumerated	How often to receive notifications
{prefix}/.subs/{id}/logs/{n}/stagger	*Unsigned*	*The window for staggered notifications*

19) **Clause W.11, 2nd §**, p. 1209,
Clause W.41.26, JSON Example, p. 1265: BACnetTrendLogRecord is an undefined type. It should be BACnetLogRecord.

W.11 Representation of Logs

. . .

There are two kinds of Logs - Trends and Events. Each record in a Trend Log is in the form of a ~~BACnetTrendLogRecord~~ *BACnetLogRecord*, even if the source of the data is not BACnet. Each record in an Event Log is in the form of a BACnetEventLogRecord, even if the source of the data is not BACnet.

...

W.41.26 Receiving a Subscription Log Callback

. . .

JSON:

```
POST /subscriber/callback/uri
Content-Type: application/json
...

{
    "$subscription": "http://theserver/.subs/4223",
    "1": {
        "$base":"List",
        "$via":"http://theserver/path/to/data/$history",
        "$memberType":~~"0-BACnetTrendLogRecord"~~ *"0-BACnetLogRecord"*,
        "543123": {
. . .
```

20) **Clause W.11.1.1, 2nd §**, p. 1210,
Clause Y.4.45, p. 1265: BACnetTrendRecord is an undefined type. It should be BACnetLogRecord.

W.11.1.1 Representation of Logs

. . .

Trend records are available with the 'history' metadata that is a List of ~~BACnetTrendRecord~~ *BACnetLogRecord* constructs.

. . .

Y.4.45 'history'

The 'history' metadata, of type List of Sequence, is the list of all trend records available for the value of the data item. Each member of the List is of type ~~BACnetTrendRecord~~ *BACnetLogRecord*. This metadata is not normally present in serialized contexts since it could be very large. This metadata only has practical use in contexts and operations (such as web services) where the representation of the contents can be limited by range selection. If the data item does not have an associated history, this metadata shall be absent, and the 'hasHistory' metadata shall be false.

21) **HISTORY OF REVISIONS**, p. 1315: The yellow highlighting of the approval dates for the revision 19 addenda 135-2012*am*, 135-2012*ba*, and 135-2012*bc* has no meaning and therefore should be removed. The current highlighting and the final non-highlighted text are shown.

1	19	**Addendum *am* to ANSI/ASHRAE 135-2012** Approved by ASHRAE on April 29, 2016; and by the American National Standards Institute on April 29, 2016. 1. . . .
1	19	**Addendum *ba* to ANSI/ASHRAE 135-2012** Approved by ASHRAE on April 29, 2016; and by the American National Standards Institute on April 29, 2016. 1. . . .
1	19	**Addendum *bc* to ANSI/ASHRAE 135-2012** Approved by ASHRAE on April 29, 2016; and by the American National Standards Institute on April 29, 2016. 1. . . .

1	19	**Addendum *am* to ANSI/ASHRAE 135-2012** Approved by ASHRAE on April 29, 2016; and by the American National Standards Institute on April 29, 2016. 1. . . .
1	19	**Addendum *ba* to ANSI/ASHRAE 135-2012** Approved by ASHRAE on April 29, 2016; and by the American National Standards Institute on April 29, 2016. 1. . . .
1	19	**Addendum *bc* to ANSI/ASHRAE 135-2012** Approved by ASHRAE on April 29, 2016; and by the American National Standards Institute on April 29, 2016. 1. . . .

22) **Clause 12.53**, p. 485: The Channel object type misses the statement on the event algorithm used for intrinsic reporting by this object.

12.53 Channel Object Type

. . .

Channel objects may optionally support intrinsic reporting to facilitate the reporting of fault conditions. Channel objects that support intrinsic reporting shall apply the NONE event algorithm.

The object and its properties are summarized in Table 12-62 and described in detail in this clause.

. . .

23) **Table 12-71**, p. 517: Several properties should be optional, and only required if Protocol_Level is BACNET_APPLICATION. See also IR 135-2016-1.

12.56 Network Port Object Type

. . .

Table 12-71. Properties of the Network Port Object Type

Property Identifier	Property Datatype	Conformance Code
. . .	. . .	. . .
Network_Number	Unsigned16	~~R[1]~~ *O*[1, 25]
Network_Number_Quality	BACnetNetworkNumberQuality	~~R~~ *O*[25]
. . .	. . .	. . .
APDU_Length	Unsigned	~~R~~ *O*[25]
. . .	. . .	. . .

[1] Required to be writable in routers, secure devices, and any other device that requires knowledge of the network number for proper operation.

. . .

[24] . . .

[25] Required to be present if Protocol_Level is BACNET_APPLICATION.

24) **Clause 12.55**, p. 508: The Binary Lighting Output object type misses the statement on the event algorithm used for intrinsic reporting by this object.

12.55 Binary Lighting Output Object Type

. . .

Binary Lighting Output objects may optionally support intrinsic reporting to facilitate the reporting of fault conditions. Binary Lighting Output objects that support intrinsic reporting shall apply the NONE event algorithm.

The object and its properties are summarized in Table 12-69 and described in detail in this clause.

. . .

25) This erratum removed June 23, 2018. Duplicate of Erratum 15)

26) **Clause 12.20, 2nd§**, p. 268: There is an extra space in CHANGE_OF _STATE that should be removed.

12.20 Multi-state Value Object Type

. . .

Multi-state Value objects that support intrinsic reporting shall apply the ~~CHANGE_OF_ STATE~~ *CHANGE_OF_STATE* event algorithm.

27) **Clause F.1.13**, p. 969: Some encodings for the ConfirmedCOVNotificationMultiple service example are incorrect. The X'27' for the 'Time Remaining' parameter does not agree with the value of 35. The binary value should be X'23'. The closing tags of the COV notification values should be X'2F', not X'2E'.

F.1.13 Encoding for Example E.1.13 - ConfirmedCOVNotificationMultiple Service

. . .	
X'29'	SD Context Tag 2 (Time Remaining, L=1)
~~X'27'~~ *X'23'*	35
. . .	
X'42820000'	65.0
~~X'2E'~~ *X'2F'*	PD Closing Tag 2 (Value)
. . .	
X'42A03333'	80.1
~~X'2E'~~ *X'2F'*	PD Closing Tag 2 (Value)
X'1F'	PD Closing Tag 1 (List of Values)
X'4F'	PD Closing Tag 4 (List of COV Notifications)

. . .

28) **Clause 13.2.2.1.4, 1st §**, p. 595: The example at the end of the first paragraph is incorrect.

13.2.2.1.4 Transition Actions

This clause describes the actions to be taken when a transition of the event-state-detection state machine occurs. The actions are the same for all transitions and they shall be executed even if the transition does not change the event state (e.g., ~~to the ToOffNormal from the OffNormal state~~ *a transition from the OFFNORMAL event state to the OFFNORMAL event state*).

29) **Clause F.1.14**, p. 969: The encoded PDU should be an unconfirmed request PDU.

F.1.14 Encoding for Example E.1.14 - UnconfirmedCOVNotificationMultiple Service

~~X'00'~~	~~PDU Type=0 (BACnet-Confirmed-Request-PDU, SEG=0, MOR=0, SA=0)~~
~~X'02'~~	~~Maximum APDU Size Accepted=206 octets~~
~~X'0F'~~	~~Invoke ID=15~~
X'10'	*PDU Type=1 (BACnet-Unconfirmed-Request-PDU)*
X'0B'	Service Choice=11 (UnconfirmedCOVNotificationMultiple-Request)

. . .

30) **Clause F.3.12**, p. 981: The description for the last byte should also indicate "TRUE".

F.3.12 Encoding for Example E.3.12 - WriteGroup Service, Example #2

. . .	
X'39'	SD Context Tag 3 (Inhibit Delay, L=1)
X'01'	1 *(TRUE)*

31) **Clause 20.2.15**, p. 777: The application tagged UTF-8 Character String example with non-ANSI characters ("Français") is misplaced. It should appear in Clause 20.2.9.

20.2.15 Encoding of a Tagged Value

. . .

~~Example: Application-tagged character string with non-ANSI character~~
~~ASN.1 = CharacterString~~
~~Value = "Français" (ISO 10646 UTF-8)~~
~~Application Tag = Character String (Tag Number = 7)~~
~~Encoded Tag = X'75'~~
~~Length Extension = X'0A'~~
~~Character Set = X'00' (ISO 10646: UTF-8)~~
~~Encoded Data = X'4672616EC3A7616973'~~

. . .

20.2.9 Encoding of a Character String Value

. . .

Example: Application-tagged character string with non-ANSI character
ASN.1 = CharacterString
Value = "Français" (ISO 10646 UTF-8)
Application Tag = Character String (Tag Number = 7)
Encoded Tag = X'75'
Length Extension = X'0A'
Character Set = X'00' (ISO 10646: UTF-8)
Encoded Data = X'4672616EC3A7616973'

In the case of IBM/Microsoft DBCS (X'01'), the initial octet shall be followed by two additional octets whose value shall

. . .

32) **Clause H.1.1.2**, p. 1003: There are two Clauses numbered H.1.1.2. The second should be H.1.1.3.

H.1.1.2 Multiple "Virtual" BACnet Devices in a Single Physical Device

. . .

~~H.1.1.2~~ *H.1.1.3* Modeling non-BACnet Data as Objects in a Single BACnet Device

. . .

33) **Clause F.4.5**, p. 983: In the title of this clause, there is a white space missing before "Service".

F.4.5 Encoding for Example E.4.5 - ~~ConfirmedTextMessageService~~ *ConfirmedTextMessage Service*

. . .

34) **Table 13-5**, p. 600: The Lighting Output object type is misspelled. There should not be a hyphen to bind the words "Lighting" and "Output".

Table 13-5. Properties Reported in CHANGE_OF_RELIABILITY Notifications

Object Type	Properties
. . .	. . .
. . . ~~Lighting-Output,~~ *Lighting Output*, . . .	Present_Value
. . .	. . .

[1] . . .

35) **Clause 12.25.10**, p. 302: The current COV subscription language for the Trend Log object should not be limited to the SubscribeCOV service. The language should be conceptual and cover any COV subscription mechanism.

12.25.10 COV_Resubscription_Interval

If the Trend Log is acquiring data from a remote device by COV subscription, this property, of type Unsigned, specifies the number of seconds between COV resubscriptions, provided that COV subscription is in effect. ~~SubscribeCOV~~ *COV subscription* requests shall specify twice this lifetime for the subscription and shall specify the issuance of confirmed notifications. If COV subscriptions are in effect, the first COV subscription is issued when the Trend Log object begins operation or when Enable becomes TRUE. If present, the value of this property shall be non-zero.

36) **Clause 6.3.2**, p. 62: ZigBee is misspelled.

6.3.2 Broadcast Messages

. . .

A local broadcast makes use of the broadcast MAC address appropriate to the local network's LAN technology, i.e. X'FFFFFFFFFFFF' for Ethernet, X'00' for ARCNET, X'FF' for MS/TP, or X'00' in the DstSubnet field of Address Format 0 in LonTalk, X'FFFF' for ~~Zigbee~~ *ZigBee*, and an IP address with all ones in the host portion for BACnet/IP.

. . .

37) **Clauses 15.8.1.1.4.3.3, 15.8.1.1.4.3.4, 15.8.1.1.4.3.5**, p. 695, 696: The time scale in the examples is erroneous.

15.8.1.1.4.3.3 Example - Positive Count

Assume a device contains a list with 1000 items and is capable of returning 200 items in a ReadRange response. The ReadRange service request contains a ‘Reference Time’ = March 18, 2013, ~~13:59:00~~ *14:19:00* and ‘Count’ = 300. The resulting ReadRange service response contains 200 items from March 18, 2013, ~~14:00:00~~ *14:20:00* to March 18, 2013,

~~17:19:00~~ *17:39:00* with FIRST_ITEM = FALSE, LAST_ITEM = FALSE, and MORE_ITEMS = TRUE. The ‘First Sequence Number’ = 2800.

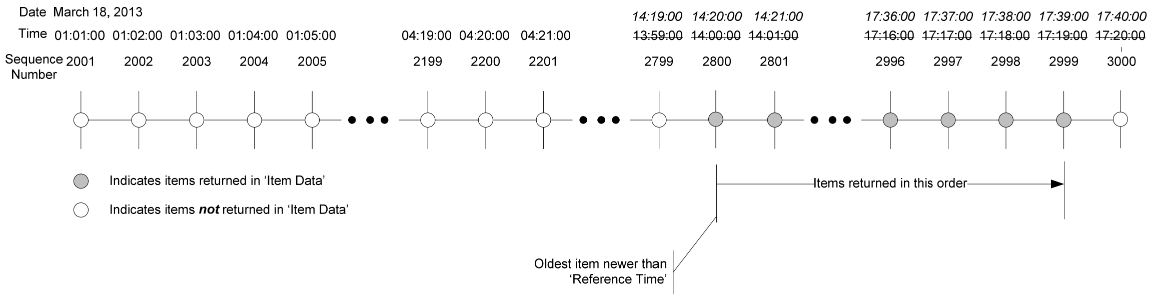
Date March 18, 2013
Time 01:01:00 01:02:00 01:03:00 01:04:00 01:05:00 04:19:00 04:20:00 04:21:00
14:19:00 *14:20:00* *14:21:00*
~~13:59:00~~ ~~14:00:00~~ ~~14:01:00~~
17:36:00 *17:37:00* *17:38:00* *17:39:00* *17:40:00*
~~17:16:00~~ ~~17:17:00~~ ~~17:18:00~~ ~~17:19:00~~ ~~17:20:00~~
Sequence Number 2001 2002 2003 2004 2005 2199 2200 2201 2799 2800 2801 2996 2997 2998 2999 3000
Indicates items returned in ‘Item Data’
Indicates items ***not*** returned in ‘Item Data’
Items returned in this order
Oldest item newer than ‘Reference Time’

Figure 15-5. By Time with a Positive Count

15.8.1.1.4.3.4 Example - Positive Count, Outdated Reference Time

Assume a device contains a list with 1000 items and is capable of returning 200 items in a ReadRange response. The ReadRange service request contains a ‘Reference Time’ = November 17, 1991, 19:20:00 and ‘Count’ = 300. The resulting ReadRange service response contains 200 items from March 18, 2013, 01:01:00 to March 18, 2013, 04:20:00 with FIRST_ITEM = TRUE, LAST_ITEM = FALSE, and MORE_ITEMS = TRUE. The ‘First Sequence Number’ = 2001.

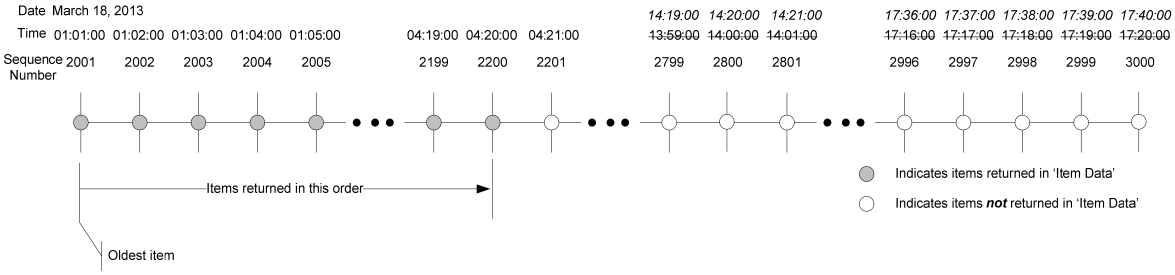
Date March 18, 2013
Time 01:01:00 01:02:00 01:03:00 01:04:00 01:05:00 04:19:00 04:20:00 04:21:00
14:19:00 *14:20:00* *14:21:00*
~~13:59:00~~ ~~14:00:00~~ ~~14:01:00~~
17:36:00 *17:37:00* *17:38:00* *17:39:00* *17:40:00*
~~17:16:00~~ ~~17:17:00~~ ~~17:18:00~~ ~~17:19:00~~ ~~17:20:00~~
Sequence Number 2001 2002 2003 2004 2005 2199 2200 2201 2799 2800 2801 2996 2997 2998 2999 3000
Items returned in this order
Oldest item
Indicates items returned in ‘Item Data’
Indicates items ***not*** returned in ‘Item Data’

Figure 15-6. By Time with a Positive Count, Outdated Reference Time

15.8.1.1.4.3.5 Example - Negative Count

Assume a device contains a list with 1000 items and is capable of returning 200 items in a ReadRange response. The ReadRange service request contains a ‘Reference Time’ = March 18, 2013, ~~17:20:00~~ *17:40:00* and ‘Count’ = -1000. The resulting ReadRange service response contains 200 items from March 18, 2013, ~~14:00:00~~ *14:20:00* to March 18, 2013, ~~17:19:00~~ *17:39:00* with FIRST_ITEM = FALSE, LAST_ITEM = FALSE, and MORE_ITEMS = TRUE. The ‘First Sequence Number’ = 2800.

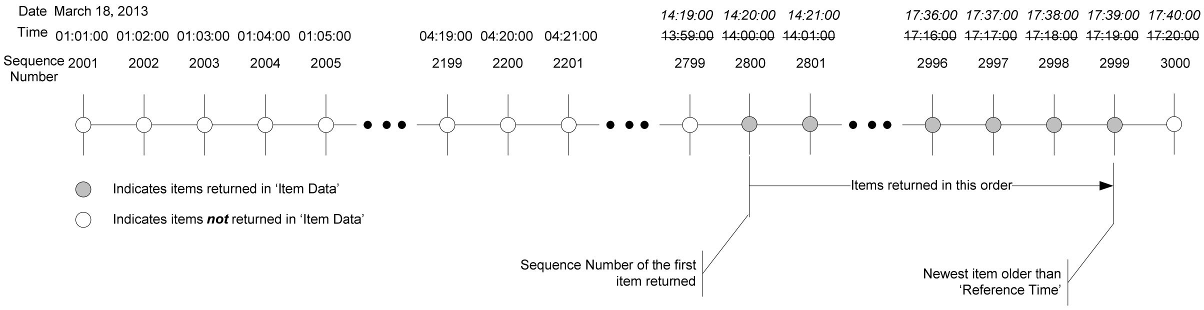
Date March 18, 2013
Time 01:01:00 01:02:00 01:03:00 01:04:00 01:05:00 04:19:00 04:20:00 04:21:00
14:19:00 *14:20:00* *14:21:00*
~~13:59:00~~ ~~14:00:00~~ ~~14:01:00~~
17:36:00 *17:37:00* *17:38:00* *17:39:00* *17:40:00*
~~17:16:00~~ ~~17:17:00~~ ~~17:18:00~~ ~~17:19:00~~ ~~17:20:00~~
Sequence Number 2001 2002 2003 2004 2005 2199 2200 2201 2799 2800 2801 2996 2997 2998 2999 3000
Indicates items returned in ‘Item Data’
Indicates items ***not*** returned in ‘Item Data’
Items returned in this order
Sequence Number of the first item returned
Newest item older than ‘Reference Time’

Figure 15-7. By Time with a Negative Count

38) **Clause 13.2.4**, p. 598: The language is on multiple services, so should be "these services".

13.2.4 Event-Summarization

. . .

Notification-servers are required to support execution of the GetEventInformation service. Support for the execution of the GetAlarmSummary and GetEnrollmentSummary services is recommended to not be implemented in devices. The specification of ~~this service~~ *these services* is retained for historical reference so that implementations of client devices have guidance on how to interoperate with older server devices.

39) **Clause 16.10.3**, p. 721,
Clause 16.10.3.1.4, p. 721,
Clause E.4.9, p. 956: The I-Am vendor identifier parameter name should be "Vendor ID", not "Vendor Identifier".

16.10.3 I-Am Service Structure

The structure of the I-Am service primitive is shown in Table 16-12. The terminology and symbology used in this table are explained in Clause 5.6.

Table 16-12. Structure of I-Am Service Primitive

Parameter Name	Req	Ind
Argument	M	M(=)
I-Am Device Identifier	M	M(=)
Max APDU Length Accepted	M	M(=)
Segmentation Supported	M	M(=)
Vendor ~~Identifier~~ *ID*	M	M(=)

16.10.3.1.4 Vendor ~~Identifier~~ *ID*

This parameter, of type Unsigned16, shall convey the identity of the vendor who manufactured the device initiating the I-Am service request. The value of this parameter shall be the same as the value of the Vendor_Identifier property of the Device object. See Clause 12.11.6 and Clause 23.

E.4.9 Examples of the Who-Is and I-Am Services

. . .

Service =	I-Am
'I-Am Device Identifier' =	(Device, Instance 3)
'Max APDU Length Accepted' =	1024
'Segmentation Supported' =	NO_SEGMENTATION
'Vendor ~~Identifier~~ *ID*' =	99

. . .

Service =	I-Am
'I-Am Device Identifier' =	(Device, Instance 1)
'Max APDU Length Accepted' =	480
'Segmentation Supported' =	SEGMENTED_TRANSMIT
'Vendor ~~Identifier~~ *ID*' =	99

Service =	I-Am
'I-Am Device Identifier' =	(Device, Instance 2)
'Max APDU Length Accepted' =	206
'Segmentation Supported' =	SEGMENTED_RECEIVE
'Vendor ~~Identifier~~ *ID*' =	33

Service =	I-Am
'I-Am Device Identifier' =	(Device, Instance 3)
'Max APDU Length Accepted' =	1024
'Segmentation Supported' =	NO_SEGMENTATION
'Vendor ~~Identifier~~ *ID*' =	99

Service =	I-Am
'I-Am Device Identifier' =	(Device, Instance 4)
'Max APDU Length Accepted' =	128
'Segmentation Supported' =	SEGMENTED_BOTH
'Vendor ~~Identifier~~ *ID*' =	66

. . .

40) **Clause 22.1.1.1**, p. 872: With the introduction of the B-GENERAL device profile, there is always a minimum device profile.

22.1.1.1 PICS Contents

. . .

(a) Basic information identifying the vendor and describing the BACnet device.
(b) The BACnet Interoperability Building Blocks supported by the device (see Annex K).
(c) The standardized BACnet device profile to which the device conforms~~, if any~~ (see Annex L).
. . .

41) **Clause 12.11.20**, p. 214: The property Max_Segments_Accepted fails to state the value requirements related to segmentation support. See interpretation requests IC135-2010-11 and IC135-2012-12.

12.11.20 Max_Segments_Accepted

The Max_Segments_Accepted property, of type Unsigned, shall indicate the maximum number of segments of an APDU that this device will accept.

If the Segmentation_Supported property of the Device object has a value of SEGMENTED_TRANSMIT or NO_SEGMENTATION, this property shall have a value of 1. If the Segmentation_Supported property of the Device object has a value of SEGMENTED_BOTH or SEGMENTED_RECEIVE, then the value of this property shall have a value greater than 1.

42) **Clause 9.5.6.5**, p. 110: The comparison of TokenCount with ($\mathbf{N_{poll}}$-1) leads to sending one token less than specified by $\mathbf{N_{poll}}$ between Poll For Master cycles. For the correct number of tokens being sent, the comparison must be with $\mathbf{N_{poll}}$.

9.5.6.5 DONE_WITH_TOKEN

. . .

SoleMaster

If FrameCount is greater than or equal to $\mathbf{N_{max\_info\_frames}}$ and TokenCount is less than $\mathbf{N_{poll}}$~~-1~~ and SoleMaster is TRUE,

then there are no other known master nodes to which the token may be sent (true master-slave operation). Set FrameCount to zero, increment TokenCount, and enter the USE_TOKEN state.

SendToken

If FrameCount is greater than or equal to $\mathbf{N_{max\_info\_frames}}$ and TokenCount is less than $\mathbf{N_{poll}}$~~-1~~ and SoleMaster is FALSE, or if NS is equal to (TS+1) modulo ($\mathbf{N_{max\_master}}$+1),

then increment TokenCount; call SendFrame to transmit a Token frame to NS; set RetryCount and EventCount to zero; and enter the PASS_TOKEN state. (The comparison of NS and TS+1 eliminates the Poll For Master if there are no addresses between TS and NS, since there is no address at which a new master node may be found in that case).

SendMaintenancePFM

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 not equal to NS,

then set PS to (PS+1) modulo ($\mathbf{N_{max\_master}}$+1); call SendFrame to transmit a Poll For Master frame to PS; set RetryCount to zero; and enter the POLL_FOR_MASTER state.

ResetMaintenancePFM

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 FALSE,

then set PS to TS; call SendFrame to transmit a Token frame to NS; set RetryCount and EventCount to zero; set TokenCount to one; and enter the PASS_TOKEN state.

SoleMasterRestartMaintenancePFM

If FrameCount is greater than or equal to $\mathbf{N_{max\_info\_frames}}$, TokenCount is greater than or equal to $\mathbf{N_{poll}}$ ~~-1~~, (PS+1) modulo ($\mathbf{N_{max\_master}}$+1) is equal to NS, and SoleMaster is TRUE,

43) **Clause 12.40.7**, p. 416: The BitString Value object language for the Status_Flags IN_ALARM bit is wrong. It cannot always be FALSE (0). This object can optionally support intrinsic alarm/event and fault reporting.

12.40.7 Status_Flags

. . .

where:

IN_ALARM ~~Always Logical FALSE (0).~~ *Logical TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).*

. . .

44) **Clause X.2**, p. 1275: In the paragraph after the bullet list, on second line, "manufacture's" is misspelled.

X.2 xdd Files

. . .

For example, an xdd file that was discovered from the "Profile_Location" of a BACnet device might contain: the PICS for that device, some links to other xdd files containing common data definitions on the ~~manufacture's~~ *manufacturer's* website, the data definitions for data specific to that device, information about the arrangement of data within the device, and links to external documentation about that device.

45) **Clause X.3**, p. 1276: "standards objects" is misspelled.

X.3 Example of Definition of Objects, Properties, and Datatypes.

Any vendor's product may contain different "flavors" of ~~standards~~ *standard* objects, proprietary extensions to standard objects, or proprietary object types, properties, and datatypes.

46) **Header Annex H**, pp. 1003: In header of this annex, there is a missing space before "WITH".

ANNEX H - COMBINING BACnet ~~NETWORKSWITH~~ *NETWORKS WITH* NON-BACnet NETWORKS (NORMATIVE)

47) **Clause Q.2.1.3**, p. 1146: In the example, second last line, the closing </TagDefinitions> is missing a slash '/'

Q.2.1.3 <TagDefinitions>

. . .

        <Description locale="tlh">...</Description>
      </String>
  ~~<TagDefinitions>~~ *</TagDefinitions>*
</CSML>

48) **Clause W.9**, p. 1208, 2nd§, 3rd line: CSML is misspelled.

W.9 Representation of Data

The 'alt' query parameter controls the format for representing data.

For the format, alt=xml, data shall be represented as the XML element corresponding to the data item's base type, such as <Real>, <Sequence>, <Array>, etc., as defined in Annex Q. The names for <Array>, <List>, and <SequenceOf> members are required in this context. The HTTP Content-Type shall be set to "application/xml". The ~~CMSL~~ *CSML* namespace shall be set as the default namespace on the topmost element. The XML format applies to GET, PUT, POST operations. Other methods shall generate a WS_ERR_BAD_METHOD error response.

. . .

49) **Clause W.24**, p. 1225, 1st §, 4th line: The word "and" should be removed.

W.24 Commandability

In addition to data that can be designated as 'writable', data can also be designated as 'commandable'. Commandable data has an associated 'priorityArray' metadata that is a 16 slot array that is compatible with the BACnet command prioritization mechanism defined by Clause 19.2. This is not limited to data originating from BACnet devices. Any data in the logical trees can be designated as commandable ~~and~~ additionally, so other protocols support a BACnet-compatible priority scheme and can thus be designated as 'commandable' in the data model.

. . .

50) **Clause W.28**, p. 1228, 1st §: The language of this paragraph misses some words.

W.28 Creating Data

If the server allows it, new members of the collection types Collection, List, SequenceOf, and Array shall be creatable by POSTing a fully formed data *item* to the path of the collection. When POSTing to an Array or SequenceOf, the newly created resource is always added to the end of the collection, increasing the size of the collection by one. When POSTing to a List or Collection, the resultant order is a local matter and might cause other members to be rearranged. If an underlying semantic for the container or a limitation of a downstream protocol prevents duplication of members (e.g., BACnet Lists), then *the* server shall respond with WS_ERR_DUPLICATES_NOT_ALLOWED if a duplicate data *item* is POSTed.

51) **Clause W.41.1**, p. 1242: There are some typos in the paragraph after the example.

W.41.1 Getting the {prefix} to Find the Server Root

. . .

In this complex example, the client finds several BACnet/WS servers when reading the /well-known/ashrae file on host.example.com. Server A is on host.example.com rooted at "/a" and uses standard ports. Server B is also on host.example.com, rooted at "/b", and also uses standard ports. Server C is on host.example.com but uses ~~nonstnadard~~ *nonstandard* ports. Server D is on another host, and uses standard ports. Server E is on another host and uses ~~non-standard~~ *nonstandard* ports.

. . .

52) **Clause W.41.7**, p. 1245: In the title, CSML is misspelled.

W.41.7 Controlling ~~CMSL~~ *CSML* Metadata with the 'metadata' Parameter

53) **Clause Y.1.4**, p. 1285, 4th §: The word "in" should be removed.

Y.1.4 Tags

. . .

A data item can have multiple tags from multiple tagging schemes applied to it. Each tag has a name that is a selection from a set of names defined by some organization. It is expected that multiple organizations will define tagging schemes, and reference to any ~~in~~ particular scheme is beyond the scope of this specification.

. . .

54) **Clause Y.4.39**: p. 1299, 1st §: There is a typo in the last sentence of the first paragraph.

Y.4.39 'href'

This optional metadata, of type String, is used to provide the URI for the remainder of a data item's value and metadata. When present, it indicates that not all of the value and metadata is present in this location and it instructs the consumer that ~~is~~ *it* has to take action to fetch the remainder of the data from the location/protocol indicated by the 'href' URI.

. . .

55) **Clause Y.5.1**, p. 1303: In the last example Boolean definition, the closing bracket ">" for "<Boolean" misses the slash '/'.

Y.5.1 'namedValues'

. . .

<Boolean name="issue-confirmed-notifications" value="true"…/>

. . .

56) **Clause Y.5.1, last §**, p. 1303: Commas are confused and the word "can" is missing:

Y.5.1 'namedValues'

. . .

Members of ~~'namedValues',~~ *'namedValues' can* have optional metadata, 'displayNameForWriting', 'notForWriting', and 'notForReading' that are available to them to provide extra information specifically for their use in the context of 'namedValues'. These metadata have no meaning outside of that context.

57) **Clause Y.12.11, 1st §**, p. 1315: The word "supports" is misspelled.

Y.12.11 BitString

BitString primitive data is modeled with the BitString base type. In addition to the common metadata described in Clause Y.4, the BitString base type also ~~support~~ *supports* the value specifiers described in Clause Y.7, the length restrictions described in clause Y.10, and the named values described in Clause Y.5.

. . .

58) **Clause Y.12.18, 1st §**, p. 1317: Remove duplicate word "that" and correct the base type.

Y.12.18 TimePattern

Time data that ~~that~~ is allowed to contain individually "unspecified" fields is modeled with the TimePattern base type. In addition to the common metadata described in Clause Y.4, the ~~Time~~ *TimePattern* base type also supports the value specifier described in Clause Y.7 and the named values described in Clause Y.5.

. . .

59) **Clause Y.13.9, 2nd §**, p. 1320: The correct term is "Standards Development Organization".

Y.13.9 Object

. . .

Object definitions are generally publically defined in some way, either through a ~~Standard Setting Organization's~~ *Standards Development Organization's* publications, or through a vendor's web site.

. . .

60) **Clause Y.13.9, last §**, p. 1320: In the last sentence, "e.g.," should be upper case.

Y.13.9 Object

. . .

The CSML type name for standard BACnet objects shall be constructed from the Clause 21 identifier in the BACnetObjectType enumeration. The CSML type name shall be "0-" plus the Clause 21 identifier with dashes removed and the initial letter of each word capitalized, plus the word "Object". ~~e.g.,~~ *E.g.,* the Clause 21 identifier "trend-log-multiple" becomes "0-TrendLogMultipleObject" as a type name.

61) **Clause Y.16.2.3, last §**, p. 1322: The word "returned" is misspelled.

Y.16.2.3 'next'

. . .

The server shall ensure that the use of the 'next' pointer functions consistently for the client. The combined results of a series of partial results using 'next' links shall be the same as if the entire result set had been ~~retuned~~ *returned* at once, with the exception that items that have been removed subsequent to the initial partial result shall not be included in future partial results. . . .

62) **Clause Y.16.2.6**, p. 1323: In the last sentence, "e.g.," should be upper case.

Y.16.2.6 'alternate'

The 'alternate' metadata, of type Link, can be applied to any data. Its value contains the URI that provides an alternate means of accessing the resource. ~~e.g.,~~ *E.g.,* if some of the data is not readable for a given authorization context, this can provide an alternate context to use.

63) **Clause Y.20.3, last §**, p. 1333: "BACnetWeekNDay" is misspelled.

Y.20.3 WeekNDay

. . .

The numeric fields do not have leading zeros. The M, W, and D fields are separated by a comma (",") character. The range and meaning of the numeric values for M, W, and D is described in the ~~BACnet WeekNDay~~ *BACnetWeekNDay* production in Clause 21.

64) **Clause Z.1.2, 1st §**, p. 1337: The reference to subsequent clauses is mistyped.

Z.1.2 Syntax Examples

Some examples using the Clause 21 datatypes will provide an introduction to the form and capabilities of the syntax. The full details of the JSON objects and members is defined ~~is subsequence~~ *in subsequent* clauses. The description of the data model and the system for defining and extending data types and expressing instances of those types is described in Annex Y. In this syntax, names of a data items are used as the JSON names; names of metadata items are prefixed with "$" and names for things that are not part of the Annex Y common data model are prefixed with "$$".

. . .

65) **Clause Z.2.4**, p. 1343: In the paragraph after the example:

Z.2.4 "$$includes"

. . .

Otherwise, the absolute or relative path is processed with respect to *the* base URI of the referring file according to RFC 3986.

. . .

66) **Table 13-1a-2**, p. 589: The footnote 1 should refer to 19.5.2, not 19.4.2.

Table 13-1a-2. Criteria for COV Reporting for Properties with Specialized Criteria.

Property	Criteria	Properties Reported
Value_Source (for commandable properties)	If criteria for the COV reporting for the object are met (as per Table 13-1) or	The values listed for the object in Table 13-1 (if present in the table, otherwise Present_Value and Status_Flags),

	Value_Source changes[1] or Current_Command_Priority changes	Value_Source, Last_Command_Time, Current_Command_Priority
Value_Source (for non-commandable properties)	If criteria for the COV reporting for the object are met (as per Table 13-1) or Value_Source changes[1]	The values listed for the object in Table 13-1 (if present in the table, otherwise Present_Value and Status_Flags), Value_Source

[1] See Clause ~~19.4.2~~ *19.5.2* for further requirements on notifications related to Value_Source changes.

67) **Clause 12.11.72**, p. 221: The reference should be to Clause Q.8.

12.11.72 Deployed_Profile_Location

This property, of type CharacterString, is the URI of the location of an xdd file (See Clause X.2). The URI is restricted to using only the "http", "https", and "bacnet" URI schemes. See ~~Clause Q~~ *Clause Q.8* for the definition of the "bacnet" URI scheme. The referenced xdd file contains additional information about the deployed device. It is intended to be used as a supplement to the information referenced by the Profile_Location property. If present, this property shall be writable and shall, at a minimum, support storage of strings with an encoded length up to 255 octets.

68) **Clause 12.56.14**, p. 527: The 'Error Code' should be VALUE_OUT_OF_RANGE, not OUT_OF_RANGE.

12.56.14 Command

This property, of type BACnetNetworkPortCommand, is used to request that the Network Port object perform various actions.

. . .

Writing a value of IDLE to this property shall result in the return of a Result(-) with an 'Error Class' of PROPERTY and an 'Error Code' of ~~OUT_OF_RANGE~~ *VALUE_OUT_OF_RANGE*.

69) **Clause 12.2.38**, p. 136, 1st paragraph, and
Clause 12, in the 1st paragraph in all following clauses for the **"Tags"** properties in all object types: The word "tag" should be in plural.

12.X.Y Tags

This property, of type BACnetARRAY of BACnetNameValue, is a collection of tags for the object. See Clause Y.1.4 for restrictions on the string values used for the names of these ~~tag~~ *tags* and for a description of tagging and the mechanism by which tags are defined.

. . .

70) **Clause 21, SubscribeCOVPropertyMultiple-Request**, p. 789: The parameter "issue-confirmed-notifications" is a required parameter, not optional.

```
SubscribeCOVPropertyMultiple-Request ::= SEQUENCE {
      subscriber-process-identifier          [0] Unsigned32,
      issue-confirmed-notifications          [1] BOOLEAN ~~OPTIONAL~~,
      lifetime                               [2] Unsigned OPTIONAL,
      max-notification-delay                 [3] Unsigned OPTIONAL,
      list-of-cov-subscription-specifications [4] SEQUENCE OF SEQUENCE {
            monitored-object-identifier             [0] BACnetObjectIdentifier,
            list-of-cov-references                  [1] SEQUENCE OF SEQUENCE {
                  monitored-property                       [0] BACnetPropertyReference,
                  cov-increment                            [1] REAL OPTIONAL,
                  timestamped                              [2] BOOLEAN
                  }
            }
      }
```

71) **Table 12-69**, p. 509,
Clause 12.55.6, p. 511, and
Clause 12.55.7, p. 512, **to Clause 12.55.38**, p. 516: The Binary Lighting Output object type failed to include the Event_State property. Note that with the insertion of the Event_State property, the Clauses for Reliability and for all subsequent properties are renumbered accordingly.

Table 12-69. Properties of the Binary Lighting Output Object Type

Property Identifier	Property Datatype	Conformance Code
. . .	. . .	. . .
Status_Flags	BACnetStatusFlags	R
Event_State	*BACnetEventState*	*O^{1}*
Reliability	BACnetReliability	O
. . .	. . .	. . .
Polarity	BACnetPolarity	O
Elapsed_Active_Time	Unsigned32	~~O^{1}~~ *O^{2}*
Time_Of_Active_Time_Reset	BACnetDateTime	~~O^{1}~~ *O^{2}*
Strike_Count	Unsigned	~~O^{2}~~ *O^{3}*
Time_Of_Strike_Count_Reset	BACnetDateTime	~~O^{2}~~ *O^{3}*
Event_Detection_Enable	BOOLEAN	~~O^{3,4}~~ *O^{1,4}*
Notification_Class	Unsigned	~~O^{3,4}~~ *O^{1,4}*
Event_Enable	BACnetEventTransitionBits	~~O^{3,4}~~ *O^{1,4}*
Acked_Transitions	BACnetEventTransitionBits	~~O^{3,4}~~ *O^{1,4}*
Notify_Type	BACnetNotifyType	~~O^{3,4}~~ *O^{1,4}*
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	~~O^{3,4}~~ *O^{1,4}*
Event_Message_Texts	BACnetARRAY[3] of CharacterString	O^{4}
. . .	. . .	. . .

~~3~~ *1* These properties are required if the object supports intrinsic reporting.

~~1~~ *2* If one of the optional properties Elapsed_Active_Time or Time_Of_Active_Time_Reset is present, then both of these properties shall be present.

~~2~~ *3* If one of the optional properties Strike_Count or Time_Of_Strike_Count_Reset is present, then both of these properties shall be present.

4 These properties shall be present only if the object supports intrinsic reporting.

. . .

12.55.6 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a Binary Lighting Output object. Two of the flags are associated with the values of other properties of this object. A more detailed status could be determined by reading the properties that are linked to these flags. The relationship between individual flags is not defined by the protocol. The four flags are

{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}

where:

IN_ALARM ~~Always Logical FALSE (0).~~ *Logical TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).*

FAULT Logical TRUE (1) if the Reliability property is present and does not have a value of NO_FAULT_DETECTED, otherwise logical FALSE (0).

OVERRIDDEN Logical TRUE (1) if the output has been overridden by some mechanism local to the BACnet device, otherwise logical FALSE (0). In this context "overridden" is taken to mean that the physical output is no longer tracking changes to the Present_Value property, and the Reliability property is no longer a reflection of the physical output.

OUT_OF_SERVICE Logical TRUE (1) if the Out_Of_Service property has a value of TRUE, otherwise logical FALSE (0).

12.55.7 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.

~~12.55.7~~ *12.55.8* Reliability

. . .

72) **Clause W.5.3**, p. 1201: The format for public keys in /.auth cannot be PKCS #8 and was missed to be defined explicitly.

W.5.3 The .auth Data Item

The .auth data item contains information related to the server device's security. The meaning of this data is discussed in Clause W.3. All data under the /.auth path, with the exception of the "{item}-pend" items, shall be nonvolatile. All certificates shall be X.509 certificates in binary DER format with a mediaType "application/x-x509-ca-cert", ~~and~~ all *private* keys shall be in PKCS #8 binary DER format (RFC 5958) with a mediaType "application/pkcs8", *and all public keys shall be a Subject Public Key Info structure as defined by X.509 in Section 4.1.2.7 of RFC 5280, in binary DER format*.

The complete list of children is defined in the following table.

. . .

73) This erratum removed June 23, 2018. Duplicate of Erratum 23)

74) **Clause 12.4.33**, p. 174: Make language consistent on where "read-only" is placed.

12.4.33 Resolution
This *read-only* property, of type REAL, indicates the smallest recognizable change in Present_Value in engineering units ~~(read-only)~~.

75) **Clause 12.3.15**, p. 166,
Clause 12.4.11, p. 172,
Clause 12.7.21, p. 192,
Clause 12.8.19, p. 199,
Clause 12.19.13, p. 264,
Clause 12.20.12, p. 271,
Clause 12.26.10, p. 311,
Clause 12.37.10, p. 399,
Clause 12.38.10, p. 405
Clause 12.39.11, p. 410
Clause 12.40.11, p. 417,
Clause 12.41.10, p. 422,
Clause 12.42.10, p. 427,
Clause 12.43.11, p. 432,
Clause 12.44.11, p. 439,
Clause 12.45.10, p. 446,
Clause 12.46.10, p. 451,
Clause 12.47.10, p. 456,
Clause 12.48.10, p. 461,
Clause 12.54.20, p. 505,
Clause 12.55.13, p. 513: Make language for the Priority_Array property consistent, in particular on where "read-only" is placed.

12.3.15 Priority_Array
12.54.20 Priority_Array
12.55.13 Priority_Array
This *read-only* property, *of type BACnetPriorityArray,* is ~~a read-only~~ *an* array ~~of prioritized values~~ *that contains prioritized commands that are in effect for this object.* See Clause 19 for a description of the prioritization mechanism.

12.4.11 Priority_Array
12.7.21 Priority Array
12.8.19 Priority Array
12.26.10 Priority_Array
This *read-only* property, *of type BACnetPriorityArray,* is ~~a read-only~~ *an* array that contains prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism.

12.19.13 Priority_Array
12.20.12 Priority_Array
This *read-only* property, *of type BACnetPriorityArray,* is ~~a read-only~~ *an* array that contains prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism. Any local modification to the values in the Priority_Array when the Number_Of_States property is changed is a local matter.

12.37.10 Priority_Array
12.38.10 Priority_Array
12.39.10 Priority_Array
12.40.11 Priority_Array
12.41.10 Priority_Array
12.42.10 Priority_Array
12.43.11 Priority_Array
12.44.11 Priority_Array
12.45.10 Priority_Array
12.46.10 Priority_Array
12.47.10 Priority_Array
12.48.10 Priority_Array

This *read-only* property, of type BACnetPriorityArray, is ~~a read-only~~ *an* array ~~containing~~ *that contains* prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism.

76) **Clause 12.57.28**, p. 551,
Clause 12.57.29, p. 551,
Clause 12.57.31, p. 551,
Clause 12.57.32, p. 551,
Clause 12.57.33, p. 551,
Clause 12.59.38, p. 566,
Clause 12.59.40, p. 566,
Clause 12.59.42, p. 566,
Clause 12.59.43, p. 567,
Clause 12.59.44, p. 567,
Clause 12.60.22, p. 573,
Clause 12.60.24, p. 573,
Clause 12.60.26, p. 573,
Clause 12.60.27, p. 574,
Clause 12.60.28, p. 574,
Clause K.2.32, p. 1061: The enumerations TO_OFFNORMAL, TO_NORMAL and TO_FAULT are misspelled in these clauses.

12.57.28 Event_Enable
12.59.38 Event_Enable
12.60.22 Event_Enable

This property, of type BACnetEventTransitionBits, shall convey three flags that separately enable and disable the distribution of ~~TO-OFFNORMAL~~ *TO_OFFNORMAL*, ~~TO-FAULT~~ *TO_FAULT*, and ~~TO-NORMAL~~ *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.

12.57.29 Acked_Transitions
12.59.40 Acked_Transitions
12.60.24 Acked_Transitions

This read-only property, of type BACnetEventTransitionBits, shall convey three flags that separately indicate the acknowledgment state for ~~TO-OFFNORMAL~~ *TO_OFFNORMAL*, ~~TO-FAULT~~ *TO_FAULT*, and ~~TO-NORMAL~~ *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.

12.57.31 Event_Time_Stamps
12.59.42 Event_Time_Stamps
12.60.26 Event_Time_Stamps
This read-only property, of type BACnetARRAY[3] of BACnetTimeStamp, shall convey the times of the last ~~TO-OFFNORMAL~~ *TO_OFFNORMAL*, ~~TO-FAULT~~ *TO_FAULT*, and ~~TO-NORMAL~~ *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.

12.57.32 Event_Message_Texts
12.59.43 Event_Message_Texts
12.60.27 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_OFFNORMAL*, ~~TO-FAULT~~ *TO_FAULT*, and ~~TO-NORMAL~~ *TO_NORMAL* events, respectively (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.

12.57.33 Event_Message_Texts_Config
12.59.44 Event_Message_Texts_Config
12.60.28 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_OFFNORMAL*, ~~TO-FAULT~~ *TO_FAULT*, and ~~TO-NORMAL~~ *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.

K.2.32 BIBB - Alarm and Event Management-Access Control-B (AE-AC-B)
. . .
Any device that supports the generation of event notifications that require operator acknowledgment shall support AE-ACK-B and AE-INFO-B. Any device that supports the generation of ~~TO-FAULT~~ *TO_FAULT* or ~~TO-OFFNORMAL~~ *TO_OFFNORMAL* event notifications shall support AE-INFO-B.

77) **Table 13-3**, p. 599: The 'Event Type' notification parameter for the acknowledgement transition (i.e. 'Notify Type' is ACK_NOTIFICATION) is mistakenly indicated to be not present.

Table 13-3. Event Notification Service Parameter Values

Service Parameter	Event State Transition (all transitions)	Acknowledgment Transition
Process Identifier	. . .	
. . .		
Event Type	When 'To State' or 'From State' is FAULT, set to CHANGE_OF_RELIABILITY, Otherwise the value associated with the event-initiating object's event algorithm.	~~Not present~~ *When 'To State' is FAULT, set to CHANGE_OF_RELIABILITY.* *When 'To State' is NORMAL, and the device can determine reporting acknowledgement of a transition from FAULT, set to CHANGE_OF_RELIABILITY.* *Otherwise the value associated with the event-initiating object's configured event algorithm.*
. . .	. . .	. . .
Notify Type	Value of Notify_Type	ACK_NOTIFICATION
AckRequired	Value of the Ack_Required bit that corresponds to 'To State'.	Not present
From State	Value of Event_State before this transition	Not present
To State	Value of property Event_State after this transition	The 'To State' parameter from the transition being acknowledged
Event Values	As defined for the Event_Type	Not present

78) **Clause W.12.3**, p. 1216: The examples miss the "$target/" prefix in the filter expression to properly express that the property of the object of the link target is applicable.

W.12.3 Filter Examples

This filter specifies that only items in the "/.data/objects" list that have a property named 'group-number' with a value of 143 will be returned.

GET /.data/objects/?filter=*$target/*group-number eq 143

This query returns all objects with a child named "status-flags" that has the "fault" bit set.

GET /.data/objects?filter=*$target/*status-flags/contains(fault)

79) **Table 12-71**, p. 516: Footnote 6 should refer to IP_DHCP_Enable, not BACnet_IP_DHCP.

Table 12-71. Properties of the Network Port Object Type

Property Identifier	Property Datatype	Conformance Code
. . .		
IP_Address	OCTET STRING	O[6]
. . .		
IP_DHCP_Enable	BOOLEAN	O[8]
. . .		
Profile_Name	CharacterString	O

. . .

[6] Required if the port is a BACnet/IP port. If the ~~BACnet_IP_DHCP~~ *IP_DHCP_Enable* property is TRUE, and this property is configured by DHCP, this property shall be read-only.

. . .

80): **Clause 12.50**, pp. 465: Global Group pSelectFlags inconsistencies with 12.50.10.

12.50 Global Group Object Type

…

Global Group objects that support intrinsic reporting shall apply the CHANGE_OF_STATUS_FLAGS event algorithm. ~~The pSelectedFlags parameter used shall only have the IN_ALARM bit set.~~

…

81) **Clause K.1.19**, pp. 1042: wrong table reference

K.1.19 BIBB - Data Sharing-Modify-A (DS-M-A)

…

A device claiming support for DS-M-A is interoperable with devices that support DS-WP-B and support one or more of the objects listed in ~~Table K-1~~*Table K5*.

82) **Clause 12.12.16**, pp. 225: inconsistent wording

12.12.16 Event_Detection_Enable

This property, of type BOOLEAN, indicates whether (TRUE) or not (FALSE) ~~intrinsic~~*event* 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.

83) **Table 13-5**, pp. 599: Remove unnecessary Schedule object entry

Table 13-5. Properties Reported in CHANGE_OF_RELIABILITY Notifications

Object Type	Properties
…	

Program	Program_State Reason_For_Halt[2,3] Description_Of_Halt[2,3]
~~Schedule~~	~~None~~
Timer	Present_Value Timer_State Update_Time[3] Last_State_Change[3] Initial_Timeout[3] Expiration_Time[3]

[1] This value may be excluded from the property-value parameter due to security requirements.

[2] This property is, or may be, from a referenced object. If the value is not known by the event-initiating object, then it shall not be included in the property-value parameter.

[3] These properties are optional and are included only if present in the object.

84) **Status_Flags and Event_State should clearly state they are read only**

These properties are derived properties so are read only. Add “read only’ to all Status_Flags and Event_State descriptions.

Change all:
12.x.y Status_Flags
This *read-only* property, of type…

12.x.y Event_State
This *read-only* property, of type…

85) **Clause 13.2.1**, pp. 591: Statement is false as of 2012aw-6

13.2.1 Event Detection and Reporting Model

…

Event-notification-distribution refers to the process involved in sending notifications of event state transitions and acknowledgment transitions using ConfirmedEventNotifications and UnconfirmedEventNotifications to a set of notification-clients and to local Event Log objects. Event-notification-distribution is provided via Notification Class objects and, optionally, Notification Forwarder objects. Devices that support event-state-detection shall support event-notification-distribution.

Objects support event-state-detection via intrinsic reporting, or event-state-detection can be provided for the object via another object performing algorithmic reporting. ~~Notification Class, Notification Forwarder and any objects that implement algorithmic reporting shall not be permitted to implement intrinsic reporting.~~

The following clauses specify the Event Detection and Reporting model independent of the object types. The Event Log and Notification Forwarder object processes notifications and are included in the overview for information. These objects and their associations to event notifications are specified in Clause 12.

86) **Clause 12.57**, pp. 541: Missing “set Present_Value to zero” for State equals RUNNING

…

State RUNNING

In the RUNNING state, the timer is active and is counting down the remaining time. The Present_Value property shall indicate the remaining time until expiration. The Timer_Running property shall have a value of TRUE. The

Expiration_Time property shall indicate the date and time when the timer will expire. The value of Expiration_Time shall be calculated at the time the property is read.

Timer Expired
If the remaining time indicated by Present_Value reaches zero,

then set Timer_Running to FALSE; set Timer_State to EXPIRED; set Last_State_Change to RUNNING_TO_EXPIRED; set Expiration_Time to the current data and time; set Update_Time to the current date and time; initiate the write requests for the RUNNING_TO_EXPIRED transition if present; and enter the EXPIRED state.

Expire Request
If a value of zero is written to the Present_Value property, or a value of FALSE is written to the Timer_Running property,

then set Timer_Running to FALSE; set Timer_State to EXPIRED; set Last_State_Change to FORCED_TO_EXPIRED; *set Present_Value to zero;* set Expiration_Time to the current date and time; set Update_Time to the current date and time; initiate the write requests for the FORCED_TO_EXPIRED transition if present; and enter the EXPIRED state.

…

87) **Clarify COV Increment in active COV subscriptions**

3.1 Terms Adopted from International Standards

…

notification-server: a BACnet device that contains event-initiating objects and performs event notification distribution.

numeric quantity*: a value that is the magnitude of a quantity. A representation of a specific amount such as temperature, volume or counts.*

object: a specific instance of an object type. While an object type is identified by a unique Object_Type property, an object is identified by its Object_Identifier property.

…

12.5.13 Object_Property_Reference

This property, of type BACnetDeviceObjectPropertyReference, shall identify the object and property whose value is to be sampled during the 'Window_Interval'. The object referenced may be located within the device containing the Averaging object, or optionally the Averaging object may support the referencing of object properties in other devices. External references may be restricted to a particular set of BACnet devices. The *value of the property referenced in the* referenced object property must *represent a numeric quantity*~~have any of the be a numeric datatypes BOOLEAN, INTEGER, Unsigned, Enumerated or REAL~~. All sampled data shall be converted to REAL for calculation purposes. BOOLEAN FALSE shall be considered to be zero and TRUE shall be considered to be one. Enumerated datatypes shall be treated as Unsigned values. If an implementation supports writing to 'Object_Property_Reference', then if 'Object_Property_Reference' is written to using BACnet services, then all of the buffer samples shall become invalid, 'Attempted_Samples' shall become zero, 'Valid_Samples' shall become zero, 'Minimum_Value' shall become INF, 'Average_Value' shall become NaN and 'Maximum_Value' shall become -INF.

12.11.39 Active_COV_Subscriptions

The Active_COV_Subscriptions property is a BACnetLIST of BACnetCOVSubscription, each of which consists of a Recipient, a Monitored Property Reference, an Issue Confirmed Notifications flag, a Time Remaining value and an optional COV Increment. This property provides a network-visible indication of those COV subscriptions that are active at any given time. Whenever a COV Subscription is created with the SubscribeCOV or SubscribeCOVProperty service, a new entry is added to the Active_COV_Subscriptions list. Similarly, whenever a COV Subscription is terminated, the corresponding entry shall be removed from the Active_COV_Subscriptions list. *If the subscribed-to property represents a numeric quantity, the COV Increment in use for the COV subscription shall be included in the Active_COV_Subscriptions entry otherwise it is a local matter whether the COV Increment is included in the Active_COV_Subscriptions entry.*

12.11.69 Active_COV_Multiple_Subscriptions

…

Whenever a COV-multiple context is created with the SubscribeCOVPropertyMultiple service, a new entry is added to the Active_COV_Multiple_Subscriptions list if no entry is present for the Recipient and form of notification. If an entry exists, COV subscription specifications are added or modified in the list of COV subscription specifications of the entry. Similarly, whenever a COV-multiple subscription is terminated, the corresponding COV subscription specifications of the entry shall be removed. If no COV subscription specifications remain in the entry, or the remaining time indicated in the entry reaches zero, the entire entry is removed from the Active_COV_Multiple_Subscriptions list. *If the subscribed-to property in a COV specification represents a numeric quantity, the COV Increment in use for the COV subscription shall be included in the Active_COV_Multiple_Subscriptions corresponding COV specification entry otherwise it is a local matter whether the COV Increment is included in the Active_COV_Multiple_Subscriptions entry.*

13.1 Change of Value Reporting

…

When a BACnet standard object, of a type listed in Table 13-1, supports COV reporting it shall support COV reporting for the property as listed in Table 13-1. At the implementor's discretion, COV reporting may also be supported for any other property of the object. For properties listed in Table 13-1 that ~~have~~*represent* a numeric *quantity*~~datatype~~, the COV increment used to determine when to generate notifications will be the COV_Increment property of the object unless a COV_Increment parameter is supplied in the SubscribeCOVProperty or SubscribeCOVPropertyMultiple service. For other properties that ~~have~~*represent* a numeric *quantity*~~datatype~~, the COV increment to use when not supplied with the SubscribeCOVProperty or SubscribeCOVPropertyMultiple service shall be a local matter. This is to allow multiple subscribers that do not require a specific increment to use a common increment to allow for the reduction of the processing burden on the COV-server. The criteria for COV reporting for properties other than those listed in Table 13-1 is based on the datatype of the property subscribed to and is described in Table 13-1a.

…

13.16.3.1.2.2 COV Increment

The 'COV Increment' parameter, of type REAL, shall specify the minimum change in the monitored property that will cause a COV notification to be queued up or issued to subscriber COV-clients. This parameter shall be ignored if the *value*~~datatype~~ of the monitored property *does not represent a numeric quantity or a BACnetARRAY of numeric quantities*~~is not~~~~REAL or array of REAL~~.

If the monitored property is Present_Value, its ~~datatype is REAL,this~~*value represents a numeric quantity*, *the 'COV Increment'* parameter is not present, and the monitored object has a COV_Increment property, then the COV increment to use is taken from the COV_Increment property of the monitored object. Otherwise, if this parameter is not present, the value used for the COV increment shall be a local matter. The intent is to allow the subscriber to use a previously established COV increment from another subscription or to allow use of the COV_Increment property in the monitored object.

Table 13-1a. Criteria Used for COV Reporting of Properties Other Than Those Listed in Table 13-1.

~~Datatype~~*Value*	Criteria	Properties Reported
~~REAL~~*A value that represents a numeric quantity*	If the *value of the* property changes by the increment (from the service if provided; otherwise, as determined by the device) or Status_Flags changes at all (if the object has a Status_Flags property)	The subscribed-to property, Status_Flags (if the object has a Status_Flags property)
All other ~~datatypes~~*values*	If the property changes at all or Status_Flags changes at all (if the object has a Status_Flags property)	The subscribed-to property, Status_Flags (if the object has a Status_Flags property)

88) **Annex J and Annex U**, Clarify language in Annex J and Annex U clauses

J.2.10 Distribute-Broadcast-To-Network: Purpose

~~This message provides a mechanism whereby a foreign device may cause a BBMD to broadcast a message on all IP subnets in the BBMD's BDT~~ *This message provides a mechanism for a foreign device to request a BBMD to distribute a Forwarded-NPDU BVLC to the local IP subnet, to all BBMD's configured in the BBMD's BDT, and to all foreign devices in the BBMD's FDT except the originating node.*

J.5.1 Foreign Device Definition

A "foreign" device is a BACnet device that has an IP subnet address different from those *subnets* comprising the BACnet/IP network that the device seeks to join. The foreign device may be a full-time node on the foreign subnet or may be a part-time participant~~, as would be the case if the device accessed the internet via a SLIP or PPP connection~~. See Figure J-3.
…

J.5.2.2 Use of the BVLL Register-Foreign-Device Message

Upon receipt of a BVLL Register-Foreign-Device message, a BBMD *configured to accept* ~~capable of providing~~ foreign device *registrations* ~~support~~ and having available table entries, shall add an entry to its FDT as described in J.5.2.1 and reply with a BVLC-Result message containing a result code of ~~X'0000'~~*'Successful completion'* indicating the successful completion of the registration. A BBMD *that does not have an available table entry or that is not configured to accept foreign device registrations* ~~incapable of providing foreign device support~~ shall return a BVLC-Result message containing a result code of ~~X'0030'~~*'Register-Foreign-Device NAK'* indicating that the registration has failed. A B/IP device which is not configured as a BBMD shall always return a BVLC-Result message containing a result code of ~~X'0030'~~*'Register-Foreign-Device NAK'* indicating that the Register-Foreign-Device BVLL message is not supported.

U.4.5.3 Use of the BVLL Register-Foreign-Device Message

Upon receipt of a BVLL Register-Foreign-Device message, a BBMD *configured to accept* ~~that has~~ foreign device registration ~~enabled~~ and having available table entries, shall add an entry to its FDT as described in Clause U.4.5.2 and reply with a BVLC-Result message containing a result code of 'Successful completion' indicating the successful completion of the registration. A BBMD *that does not have an available table entry or that is not configured to accept foreign device registrations* ~~incapable of providing foreign device support~~ shall return a BVLC-Result message containing a result code of 'Register-Foreign-Device NAK' indicating that the registration has failed.

Upon receipt of a BVLL Register-Foreign-Device message, a BACnet/IPv6 device that is not configured as a BBMD shall return a BVLC-Result message containing a result code of 'Register-Foreign-Device NAK' indicating that the registration has failed.

89) **Clause 12.52**, pp. 479: Alert Enrollment Object Missing Event_Enable Clause

12.52.x 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.

90) **COV_Increment Guidance**

Change footnote 2 in Table 12-2. Properties of the Analog Input Object Type

Change footnote 1 in Table 12-3. Properties of the Analog Output Object Type
Change footnote 2 in Table 12-4. Properties of the Analog Value Object Type
Change footnote 4 in Table 12-20. Properties of the Loop Object Type
Change footnote 2 in Table 12-27. Properties of the Pulse Converter Object Type
Change footnote 3 in Table 12-46. Properties of the Large Analog Value Object Type
Change footnote 3 in Table 12-50. Properties of the Integer Value Object TypeChange footnote 2 in
Change footnote 3 in Table 12-51. Properties of the Positive Integer Value Object TypeChange footnote 2 in
Change footnote 2 in Table 12-64. Properties of the Lighting Output Object TypeChange footnote 2 in

Footnote: “This property is required if, and shall be present only if, the object supports COV reporting *for the Present_Value property*.”

91) **Annex T.2**, pp. 1171: Correcting an Error in the COBS Code Implementation

T.2 Decoding an Extended MS/TP Frame upon Reception

…

```
size_t
cobs_decode (uint8_t *to, const uint8_t *from, size_t length, uint8_t mask)
{
  size_t read_index = 0;
  size_t write_index = 0;
  uint8_t code, last_code;

  while (read_index < length) {
    code = from[read_index] ^ mask;
    last_code = code;
    /*
     * Sanity check the encoding to prevent the while() loop below
     * from overrunning the output buffer.
     */
    if (code == 0 || read_index + code > length) {
```

…

Erratum 92) K.5.17 – Should exclude the ACTIVATE_CHANGES choice from ReinitializeDevice service requirement

…

Devices claiming conformance to DM-BR-A are required to support *COLDSTART*, *WARMSTART*, *STARTBACKUP*, *ENDBACKUP*, *STARTRESTORE, ENDRESTORE, and ABORTRESTORE*~~all service~~ choices of the ReinitializeDevice service. In addition, devises claiming conformance to DM-BR-A shall support the device A capabilities as described in Clause 19.1.

Erratum 93) K.5.18 – Should exclude the ACTIVATE_CHANGES choice from ReinitializeDevice service requirement

…

Devices claiming conformance to DM-BR-B are required to support *COLDSTART*, *WARMSTART*, *STARTBACKUP*, *ENDBACKUP*, *STARTRESTORE, ENDRESTORE, and ABORTRESTORE*~~all service~~ choices of the ReinitializeDevice service. In addition, devices claiming conformance to DM-BR-B shall support the device B capabilities as described in Clause 19.1. Once a Restore procedure has been initiated on the device, the Read_Only property of configuration File objects shall contain the value FALSE and the File_Size property of the configuration File objects shall be writable if the size of the configuration file can change based on the device's configuration.

…

Erratum 94) K.2.7 – Missing ‘to’

…

Device B provides summaries of alarms *to* device A.

…