ADDENDA

ANSI/ASHRAE Addendum bg to
ANSI/ASHRAE Standard 135-2012

BACnet™ Data Communication Protocol for Building Automation and Control Networks

Approved by ASHRAE on February 29, 2016, and by the American National Standards Institute on March 1, 2016.

This addendum was approved by a Standing Standard Project Committee (SSPC) for which the Standards Committee has established a documented program for regular publication of addenda or revisions, including procedures for timely, documented, consensus action on requests for change to any part of the standard. The change submittal form, instructions, and deadlines may be obtained in electronic form from the ASHRAE website (www.ashrae.org) or in paper form from the Senior Manager of Standards.

The latest edition of an ASHRAE Standard may be purchased on the ASHRAE website (www.ashrae.org) or from ASHRAE Customer Service, 1791 Tullie Circle, NE, Atlanta, GA 30329-2305. E-mail: orders@ashrae.org. Fax: 678-539-2129. Telephone: 404-636-8400 (worldwide), or toll free 1-800-527-4723 (for orders in US and Canada). For reprint permission, go to www.ashrae.org/permissions.

© 2016 ASHRAE ISSN 1041-2336

Approved American National Standard
ANSI
www.ansi.org

ASHRAE Standing Standard Project Committee 135
Cognizant TC: 1.4, Control Theory and Application
SPLS Liaison: Mark P. Modera

Carl Neilson*, *Chair*	Stuart G. Donaldson*	Thomas Kurowski*
Bernhard Isler, *Vice-Chair*	Michael P. Graham*	H. Michael Newman*
Michael Osborne*, *Secretary*	David G. Holmberg*	Duffy O'Craven*
Coleman L. Brumley, Jr.*	Daniel Kollodge*	Gregory M. Spiro*
Clifford H. Copass*	Jake Kopocis*	Grant N. Wichenko*

** Denotes members of voting status when the document was approved for publication*

ASHRAE STANDARDS COMMITTEE 2015–2016

Douglass T. Reindl, *Chair*	Steven J. Emmerich	Heather L. Platt
Rita M. Harrold, *Vice-Chair*	Julie M. Ferguson	David Robin
James D. Aswegan	Walter T. Grondzik	Peter Simmonds
Niels Bidstrup	Roger L. Hedrick	Dennis A. Stanke
Donald M. Brundage	Srinivas Katipamula	Wayne H. Stoppelmoor, Jr.
John A. Clark	Rick A. Larson	Jack H. Zarour
Waller S. Clements	Lawrence C. Markel	Julia A. Keen, *BOD ExO*
John F. Dunlap	Arsen K. Melikov	James K. Vallort, *CO*
James W. Earley, Jr.	Mark P. Modera	
Keith I. Emerson	Cyrus H. Nasseri	

Stephanie C. Reiniche, *Senior Manager of Standards*

SPECIAL NOTE

This American National Standard (ANS) is a national voluntary consensus Standard developed under the auspices of ASHRAE. *Consensus* is defined by the American National Standards Institute (ANSI), of which ASHRAE is a member and which has approved this Standard as an ANS, as "substantial agreement reached by directly and materially affected interest categories. This signifies the concurrence of more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that an effort be made toward their resolution." Compliance with this Standard is voluntary until and unless a legal jurisdiction makes compliance mandatory through legislation.

ASHRAE obtains consensus through participation of its national and international members, associated societies, and public review.

ASHRAE Standards are prepared by a Project Committee appointed specifically for the purpose of writing the Standard. The Project Committee Chair and Vice-Chair must be members of ASHRAE; while other committee members may or may not be ASHRAE members, all must be technically qualified in the subject area of the Standard. Every effort is made to balance the concerned interests on all Project Committees.

The Senior Manager of Standards of ASHRAE should be contacted for

a. interpretation of the contents of this Standard,
b. participation in the next review of the Standard,
c. offering constructive criticism for improving the Standard, or
d. permission to reprint portions of the Standard.

DISCLAIMER

ASHRAE uses its best efforts to promulgate Standards and Guidelines for the benefit of the public in light of available information and accepted industry practices. However, ASHRAE does not guarantee, certify, or assure the safety or performance of any products, components, or systems tested, installed, or operated in accordance with ASHRAE's Standards or Guidelines or that any tests conducted under its Standards or Guidelines will be nonhazardous or free from risk.

ASHRAE INDUSTRIAL ADVERTISING POLICY ON STANDARDS

ASHRAE Standards and Guidelines are established to assist industry and the public by offering a uniform method of testing for rating purposes, by suggesting safe practices in designing and installing equipment, by providing proper definitions of this equipment, and by providing other information that may serve to guide the industry. The creation of ASHRAE Standards and Guidelines is determined by the need for them, and conformance to them is completely voluntary.

In referring to this Standard or Guideline and in marking of equipment and in advertising, no claim shall be made, either stated or implied, that the product has been approved by ASHRAE.

ASHRAE is a registered trademark of the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.
ANSI is a registered trademark of the American National Standards Institute.

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution,
or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

[This foreword and the "rationales" on the following pages are not part of this standard. They are merely informative and do not contain requirements necessary for conformance to the standard.]

FOREWORD

The purpose of this addendum is to present a proposed change for public review. These modifications are the result of change proposals made pursuant to the ASHRAE continuous maintenance procedures and of deliberations within Standing Standard Project Committee 135. The proposed changes are summarized below.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2012 and Addenda is indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are proposed to be added, plain type is used throughout. Only this new and deleted text is open to comment as this time. All other material in this addendum is provided for context only and is not open for public review comment except as it relates to the proposed changes.

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution,
or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135-2012*bg*-1. Add engineering units.

Rationale

Add in units for flow meters and more radiation units.

[Change **BACnetEngineeringUnits** in **Clause 21**, p. 669]
[Note: Addendum 135-2012*ar* reserved the range 47808 to 49999 for ASHRAE]

```
BACnetEngineeringUnits ::= ENUMERATED {  -- See below for numerical order
...

--Volumetric Flow
    cubic-feet-per-second                                  (142),
    cubic-feet-per-minute                                  (84),
    million-standard-cubic-feet-per-minute         (254),
    cubic-feet-per-hour                                    (191),
    standard-cubic-feet-per-day                    (47808),
    million-standard-cubic-feet-per-day            (47809),
    thousand-cubic-feet-per-day                    (47810),
    thousand-standard-cubic-feet-per-day           (47811),
    pounds-mass-per-day                                    (47812),
    cubic-meters-per-second                        (85),
    ...


-- Other
    ...
    sieverts                                                   (228),
    millisieverts                                              (229),
    microsieverts                                              (230),
    microsieverts-per-hour                         (231),
    millirems                                                  (47814),
    millirems-per-hour                                     (47815),
    decibels-a                                                 (232),
    ...
-- Numerical Order Reference
    ...
    -- see minutes-per-degree-kelvin               (236),
    ...
    -- see million-standard-cubic-feet-per-minute (254),
    -- see standard-cubic-feet-per-day             (47808),
    -- see million-standard-cubic-feet-per-day  (47809),
    -- see thousand-cubic-feet-per-day             (47810),
    -- see thousand-standard-cubic-feet-per-day (47811),
    -- see pounds-mass-per-day                         (47812),
    -- see millirems                                           (47814),
    -- see millirems-per-hour                          (47815),
    ...
    }
```

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution,
or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135-2012*bg*-2. Harmonize message text handling for all alarm services.

Rationale

Harmonize the requirement to not drop event messages for both Confirmed and Unconfirmed Event Notification and Acknowledge Alarm services when the included message text is in a character set which is not supported by the device.

[Change **Clause 13.5.2**, p. 510]
[AcknowledgeAlarm service]

13.5.2 Service Procedure

After verifying the validity of the request, the responding BACnet-user shall attempt to locate the specified object. If the object exists and if the 'Time Stamp' parameter matches the most recent time for the event being acknowledged, then the bit in the Acked_Transitions property of the object that corresponds to the value of the 'Event State Acknowledged' parameter shall be set to 1, a 'Result(+)' primitive shall be issued, and an event notification with a 'Notify Type' parameter equal to ACK_NOTIFICATION shall be issued. Otherwise, a 'Result(-)' primitive shall be issued. An acknowledgment notification shall use the same type of service (confirmed or unconfirmed) directed to the same recipients to which a confirmed or unconfirmed event notification for the same transition type would be sent. The Time Stamp conveyed in the acknowledgment notification shall not be derived from the Time Stamp of the original event notification, but rather the time at which the acknowledgment notification is generated.

A device shall *neither*~~not~~ fail to process, *nor*~~or~~ issue a Result(-), *due to* ~~upon receiving~~ an AcknowledgeAlarm service request containing an 'Acknowledgment Source' parameter in an unsupported character set. In this case, it is a local matter whether the 'Acknowledgment Source' parameter is used as provided or whether a character string, in a supported character set, of length 0 is used in its place.

[Change **Clause 13.8.2**, p. 516]
[ConfirmedEventNotification service]

13.8.2 Service Procedure

After verifying the validity of the request, the responding BACnet-user shall take whatever local actions have been assigned to the indicated event occurrence and issue a 'Result(+)' service primitive. If the service request cannot be executed, a 'Result(-)' service primitive shall be issued indicating the encountered error. A device shall *neither*~~not~~ fail to process, *nor*~~or~~ issue a Result(-), ~~upon receiving~~ *due to* a ConfirmedEventNotification service request containing a 'Message Text' parameter in an unsupported character set. *In this case, the consumption of the 'Message Text' parameter is a local matter.*

[Change **Clause 13.9.2**, p. 518]
[UnconfirmedEventNotification service]

13.9.2 Service Procedure

Since this is an unconfirmed service, no response primitives are expected. Actions taken in response to this notification are a local matter. *A device shall not fail to process a request due to a 'Message Text' parameter in an unsupported character set. In this case, the consumption of the 'Message Text' parameter is a local matter.*

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution,
or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135-2012*bg*-3. Ensure Alert Enrollment objects do not send notifications which require acknowledgment.

Rationale

Alert Enrollment objects never generate notifications which require operator acknowledgement. This change clarifies and enforces this rule regardless of the settings found in the related Notification Class object.

[Change **Clause 12.21.7**, p. 258]
[Notification Class Object]

12.21.7 Ack_Required

This property, of type BACnetEventTransitionBits, shall convey three separate flags that represent whether acknowledgment shall be required in notifications generated for TO_OFFNORMAL, TO_FAULT, and TO_NORMAL event transitions, respectively, *except in the case of the Access Point or Alert Enrollment object types which behave as directed in Clauses 12.31.40 and 12.52.8*.

[Change **Clause 12.52.8**, p. 436]
[Alert Enrollment object]

12.52.8 Notification_Class

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

The TO_NORMAL transition in the Ack_Required property of the referenced Notification Class object is ignored and the value FALSE is conveyed in the 'AckRequired' parameter of the ConfirmedEventNotification or UnconfirmedEventNotification message.

[Change **Clause 13.2.3**, p. 469]

13.2.3 Alarm-Acknowledgment

...

Whether or not an acknowledgment is required is determined by the Ack_Required property from the referenced Notification Class object, *except in the specific cases where the Ack_Required property is ignored (see Clauses 12.31.40 and 12.52.8).*

...

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution,
or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135-2012*bg*-4. Allow selection of the Nth last day of the month in a BACnetWeekNDay.

Rationale

The BACnetWeekNDay production does not currently allow for selection of "the 2nd last Thursday", or the "3rd last Friday".

[Change **Production BACnetWeekNDay** in **Clause 21**, p. 713]

BACnetWeekNDay ::= OCTET STRING (SIZE (3))
-- first octet month (1..14) 1 =January
-- 13 = odd months
-- 14 = even months
-- X'FF' = any
month
-- second octet weekOfMonth*(1..9)* where: 1 = days numbered 1-7
-- 2 = days
numbered 8-14
-- 3 = days
numbered 15-21
-- 4 = days
numbered 22-28
-- 5 = days
numbered 29-31
-- 6 = last 7 days of
this month
-- *7 = any of the 7*
days prior to the last 7 days of this month
-- *8 = any of the 7*
days prior to the last 14 days of this month
-- *9 = any of the 7*
days prior to the last 21 days of this month
-- X'FF' = any week
of this month
-- third octet dayOfWeek (1..7) where 1 = Monday
-- 7 = Sunday
-- X'FF' = any day
of week

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution,
or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135-2012*bg*-5. Remove initiation of GetEnrollmentSummary from AE-AS-A.

Rationale

Remove the requirement for initiation of GetEnrollmentSummary from the BIBB **Alarm and Event Management - Alarm Summary View – A (AE-AS-A)**. Devices which support generation of alarms are required to either support GetAlarmSummary (Protocol Revision 1 and prior) or GetEventInformation (Protocol Revision 2 and later) so the A-side device need not be capable of initiating GetEnrollmentSummary.

[Change **Clause K.2.18**, p. 888]

K.2.18 BIBB - Alarm and Event Management - Alarm Summary View - A (AE-AS-A)

Device A presents alarm summary information to the user. Device A uses GetEventInformation to retrieve or update alarm summary information presented to the user. When confronted with a device that does not support execution of GetEventInformation, Device A uses GetAlarmSummary instead. Device A may use alternate alarm and event summary services where support for execution of the alternate service is supported by Device B.

BACnet Service	Initiate	Execute
GetEventInformation	x	
~~GetEnrollmentSummary~~	~~x~~	
GetAlarmSummary	x	

Device A is not required to rely solely on the event summary services for retrieval of event information. It may use the information contained in received event notifications to build the alarm summary. In such a case, a device claiming conformance to this BIBB shall use the summarization services to update this information. Presentation content and format is a local matter.

A device claiming support for AE-AS-A is interoperable with devices that support AE-INFO-B~~, AE-ESUM-B~~ or AE-ASUM-B.

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution,
or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135-2012*bg*-6. Ensure UTC_Offset is configurable.

Rationale

Devices that use UTC need to be able to be deployed in any timezone. The range of UTC_Offset was increased to account for new time zone regions that have been created, or may be created in the future.

[Change **12.11.25**, p. 202]

12.11.25 UTC_Offset

The UTC_Offset property, of type INTEGER, shall indicate the number of minutes ~~(-780 to +780)~~ *(-1440 to +1440)* offset between local standard time and Universal Time Coordinated. The time zones to the west of the zero degree meridian shall be positive values, and those to the east shall be negative values. The value of the UTC_Offset property is subtracted from the UTC received in UTCTimeSynchronization service requests to calculate the correct local standard time. *UTC_Offset shall be configurable and accept any multiple of 15 minutes across the full range.*

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution,
or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135-2012*bg*-7. Clarify ReadRange.

Rationale

The language for the ReadRange service is unclear. These changes improve the existing requirement descriptions and provide examples for more clarity.

[Change **Clause 15.8.1.1.4**, p. 554]

15.8.1.1.4 Range

This optional parameter shall convey criteria for the consecutive range items within the referenced property that are to be returned, as described in Clause 15.8.2. The 'Range' parameter is shown in Table 15-14. The terminology and symbology used in this table are explained in Clause 5.6.

Table 15-14. Structure of the 'Range' Parameter

Parameter Name	Req	Ind	Datatype
By Position	S	S(=)	
Reference Index	M	M(=)	Unsigned
Count	M	M(=)	INTEGER16
By Sequence Number	S	S(=)	
Reference Sequence Number	M	M(=)	Unsigned32
Count	M	M(=)	INTEGER16
By Time	S	S(=)	
Reference Time	M	M(=)	BACnetDateTime
Count	M	M(=)	INTEGER16

15.8.1.1.4.1 By Position

The 'By Position' parameter shall indicate that the particular items to be read are referenced by an index.

15.8.1.1.4.1.1 Reference Index

The 'Reference Index' parameter specifies the index of the first (if 'Count' is positive) or last (if 'Count' is negative) item to be read. *If the item with the index specified in this parameter does not exist, then no items match the criteria for being read and returned, regardless of the value of the 'Count' parameter.*

15.8.1.1.4.1.2 Count

The absolute value of the 'Count' parameter specifies the number of records to be read. If 'Count' is positive, the record specified by 'Reference Index' shall be the first record read and returned; if 'Count' is negative the record specified by 'Reference Index' shall be the last record. 'Count' may not be zero.

15.8.1.1.4.1.3 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 Index' = 800 and 'Count' = 300. The resulting ReadRange service response contains 200 items from Index 800 to 999 with FIRST_ITEM = FALSE, LAST_ITEM = FALSE, and MORE_ITEMS = TRUE. The 'First Sequence Number' parameter is not included in the response.

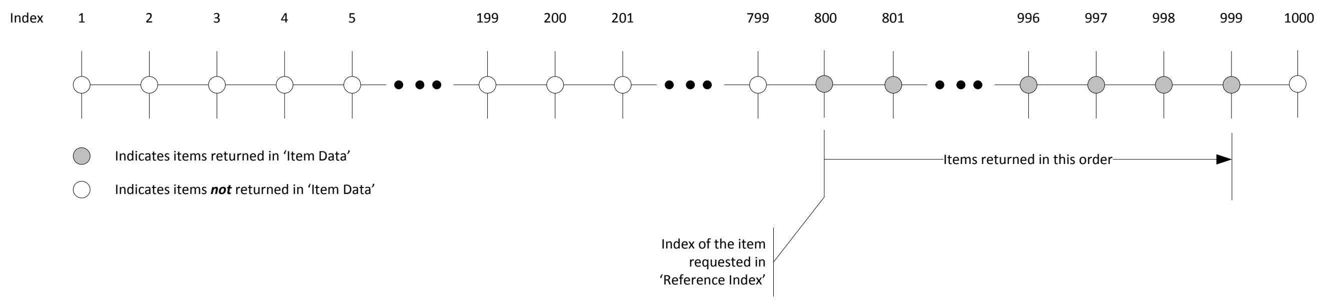

Index
1
2
3
4
5
199
200
201
799
800
801
996
997
998
999
1000
Indicates items returned in 'Item Data'
Indicates items *not* returned in 'Item Data'
Items returned in this order
Index of the item requested in 'Reference Index'

***Figure 15-1.** By Position with a Positive Count*

15.8.1.1.4.1.4 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 Index' = 1000 and 'Count' = -1000. The resulting ReadRange service

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution,
or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

response contains 200 items from Index 801 to 1000 with FIRST_ITEM = FALSE, LAST_ITEM = TRUE and MORE_ITEMS = TRUE. The 'First Sequence Number' shall not exist in the response.

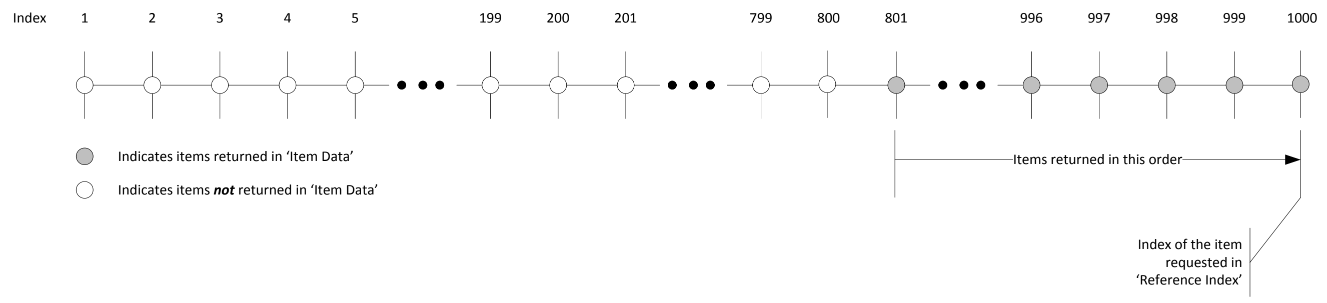

Index
1
2
3
4
5
199
200
201
799
800
801
996
997
998
999
1000
Indicates items returned in 'Item Data'
Indicates items *not* returned in 'Item Data'
Items returned in this order
Index of the item requested in 'Reference Index'

Figure 15-2. *By Position with a Negative Count*

15.8.1.1.4.2 By Sequence Number

The 'By Sequence Number' parameter shall indicate that the particular items to be read are referenced by a sequence number and that the response shall include the sequence number of the first returned item. This differs semantically from the 'By Position' parameter choice. The Reference ~~Number~~ *Index* provided in the 'By Position' choice references an item by its position in the list. In contrast, the Reference *Sequence* Number provided in the 'By Sequence Number' choice references an item by its sequence number, which it is given when the item is added to the list. Not all lists implement the concept of a sequence number. An example of a list that does implement the concept of a sequence number is the Log_Buffer property of the Trend Log object.

15.8.1.1.4.2.1 Reference Sequence Number

The 'Reference Sequence Number' parameter specifies the sequence number of the first (if 'Count' is positive) or last (if 'Count' is negative) item to be read. *If the item with the sequence number specified in this parameter does not exist, then no items match the criteria for being read and returned, regardless of the value of the 'Count' parameter.*

15.8.1.1.4.2.2 Count

The absolute value of the 'Count' parameter specifies the number of records to be read. If 'Count' is positive, the record specified by 'Reference Sequence Number' shall be the first and oldest record read and returned. If 'Count' is negative the record specified by 'Reference Sequence Number' shall be the last and newest record read and returned. 'Count' shall not be zero.

15.8.1.1.4.2.3 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 Sequence Number' = 2800 and 'Count' = 300. The resulting ReadRange service response contains 200 items from Sequence Number 2800 to 2999 with FIRST_ITEM = FALSE, LAST_ITEM = FALSE and MORE_ITEMS = TRUE. The 'First Sequence Number' = 2800.

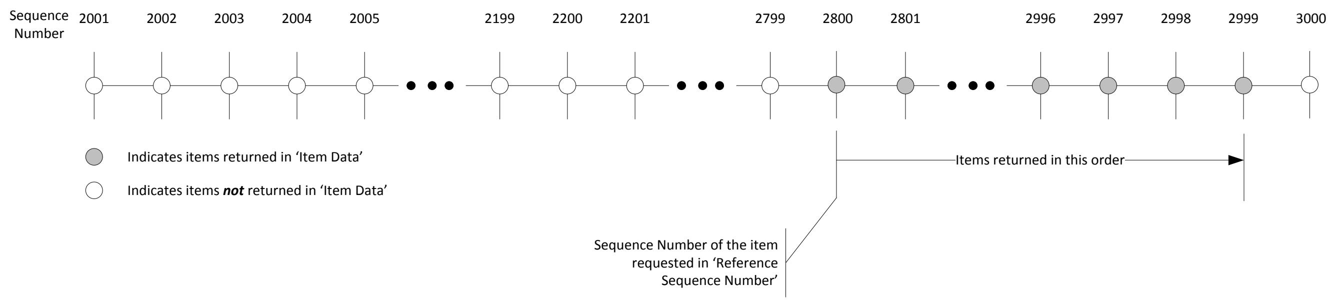

Sequence Number
2001
2002
2003
2004
2005
2199
2200
2201
2799
2800
2801
2996
2997
2998
2999
3000
Indicates items returned in 'Item Data'
Indicates items *not* returned in 'Item Data'
Items returned in this order
Sequence Number of the item requested in 'Reference Sequence Number'

Figure 15-3. *By Sequence Number with a Positive Count*

15.8.1.1.4.2.5 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 Sequence Number' = 3000 and 'Count' = -1000. The resulting ReadRange service response contains 200 items from Sequence Number 2801 to 3000 with FIRST_ITEM = FALSE, LAST_ITEM = TRUE, and MORE_ITEMS = TRUE. The 'First Sequence Number' = 2801.

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

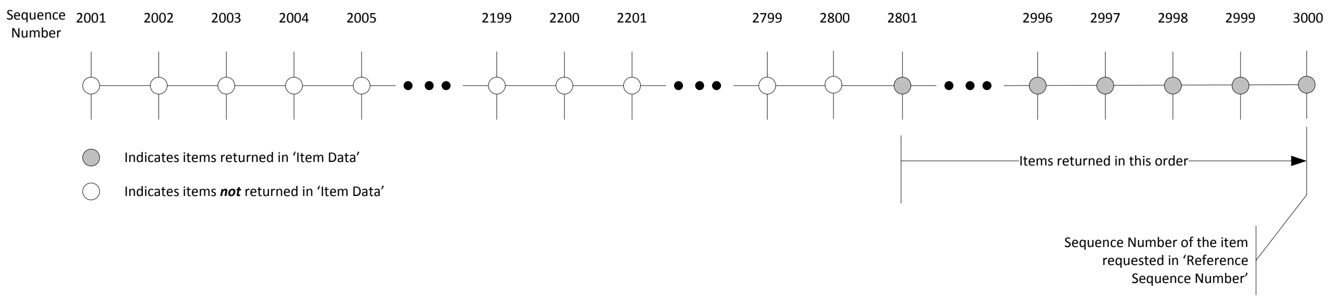

Sequence Number
2001
2002
2003
2004
2005
2199
2200
2201
2799
2800
2801
2996
2997
2998
2999
3000
Indicates items returned in 'Item Data'
Indicates items *not* returned in 'Item Data'
Items returned in this order
Sequence Number of the item requested in 'Reference Sequence Number'

Figure 15-4. *By Sequence Number with a Negative Count*

15.8.1.1.4.3 By Time

The 'By Time' parameter shall indicate that the particular item to be read is referenced by timestamp and that the Sequence Number of the item shall be returned in the response. This form of the service is expected to be used when searching lists that are loosely indexed by time.

15.8.1.1.4.3.1 Reference Time

If 'Count' is positive, the first record to be read shall be the first record with a timestamp newer than the time specified by the 'Reference Time' parameter. If 'Count' is negative, the last record to be read shall be the newest record with a timestamp older than the time specified by the 'Reference Time' parameter. This parameter shall contain a specific datetime value.

15.8.1.1.4.3.2 Count

The absolute value of the 'Count' parameter specifies the number of records to be read. If 'Count' is positive, the first record with a timestamp newer than the time specified by 'Reference Time' shall be the first and oldest record read and returned; if 'Count' is negative, the newest record with a timestamp older than the time specified by 'Reference Time' shall be the last and newest record. 'Count' shall not be zero.

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 and 'Count' = 300. The resulting ReadRange service response contains 200 items from March 18, 2013, 14:00:00 to March 18, 2013, 17:19:00 with FIRST_ITEM = FALSE, LAST_ITEM = FALSE, and MORE_ITEMS = TRUE. The 'First Sequence Number' = 2800.

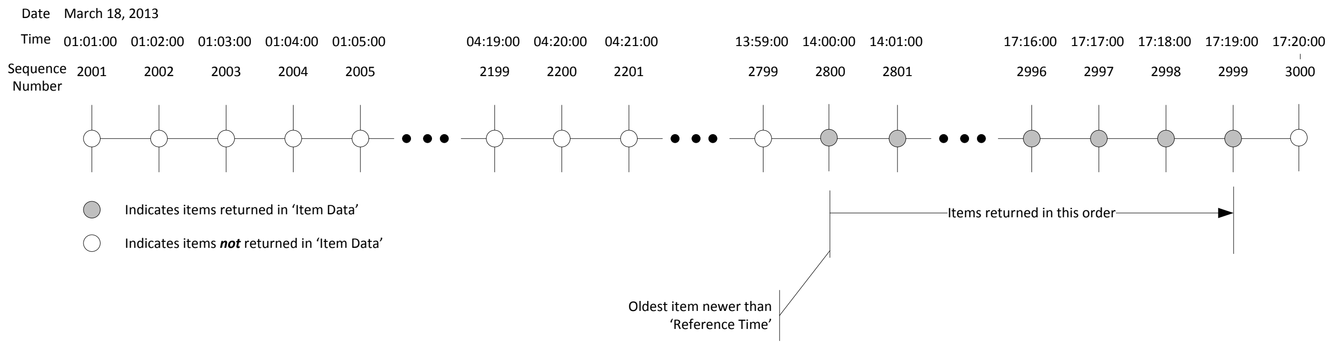

Date March 18, 2013
Time 01:01:00 01:02:00 01:03:00 01:04:00 01:05:00 04:19:00 04:20:00 04:21:00 13:59:00 14:00:00 14:01:00 17:16:00 17:17:00 17:18:00 17:19:00 17:20:00
Sequence Number 2001 2002 2003 2004 2005 2199 2200 2201 2799 2800 2801 2996 2997 2998 2999 3000
Indicates items returned in 'Item Data'
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.

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

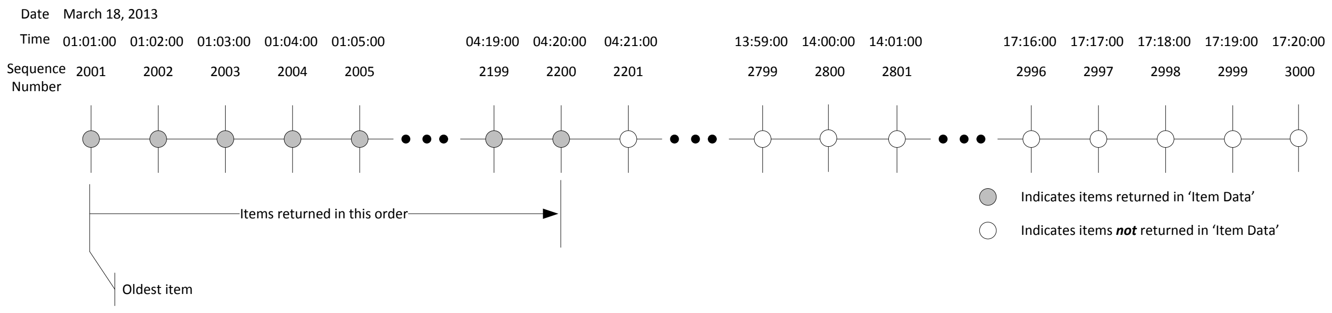

Date March 18, 2013
Time 01:01:00 01:02:00 01:03:00 01:04:00 01:05:00 04:19:00 04:20:00 04:21:00 13:59:00 14:00:00 14:01:00 17:16:00 17:17:00 17:18:00 17:19:00 17:20:00
Sequence Number 2001 2002 2003 2004 2005 2199 2200 2201 2799 2800 2801 2996 2997 2998 2999 3000
Items returned in this order
Indicates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

***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 and 'Count' = -1000. The resulting ReadRange service response contains 200 items from March 18, 2013, 14:00:00 to March 18, 2013, 17:19:00 with FIRST_ITEM = FALSE, LAST_ITEM = FALSE, and MORE_ITEMS = TRUE. The 'First Sequence Number' = 2800.

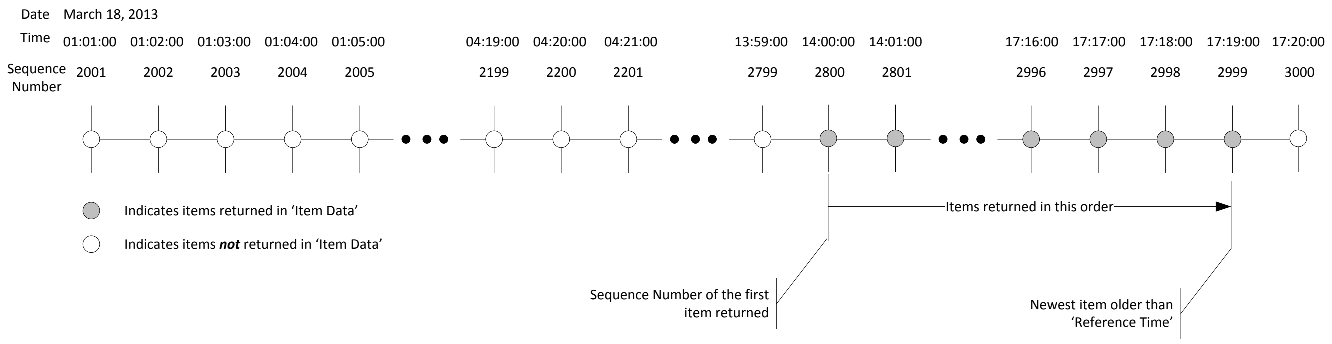

Date March 18, 2013
Time 01:01:00 01:02:00 01:03:00 01:04:00 01:05:00 04:19:00 04:20:00 04:21:00 13:59:00 14:00:00 14:01:00 17:16:00 17:17:00 17:18:00 17:19:00 17:20:00
Sequence Number 2001 2002 2003 2004 2005 2199 2200 2201 2799 2800 2801 2996 2997 2998 2999 3000
Indicates items returned in 'Item Data'
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*

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution,
or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135-2012*bg*-8. Clarify the effect of changing Buffer_Size.

Rationale

Add language to clarify the requirements when a buffer is purged due to a change to the Buffer_Size.

[Change **Clause 12.25.13**, p. 284]

12.25.13 Buffer_Size

This property, of type Unsigned32, shall specify the maximum number of records the buffer may hold. If writable, it may not be written when Enable is TRUE. The disposition of existing records when Buffer_Size is written is a local matter. *If all records are deleted when the Buffer_Size is written then the object shall act as if the Record_Count was set to zero.*

[Change **Clause 12.27.12**, p. 301]

12.27.12 Buffer_Size

This property, of type Unsigned32, shall specify the maximum number of records the buffer may hold. If writable, it may not be written when Enable is TRUE. The disposition of existing records when Buffer_Size is written is a local matter. *If all records are deleted when the Buffer_Size is written then the object shall act as if the Record_Count was set to zero.*

[Change **Clause 12.30.18**, p. 322]

12.30.18 Buffer_Size

This property, of type Unsigned32, shall specify the maximum number of records the buffer may hold. If writable, it may not be written when Enable is TRUE. The disposition of existing records when Buffer_Size is written is a local matter. *If all records are deleted when the Buffer_Size is written then the object shall act as if the Record_Count was set to zero.*

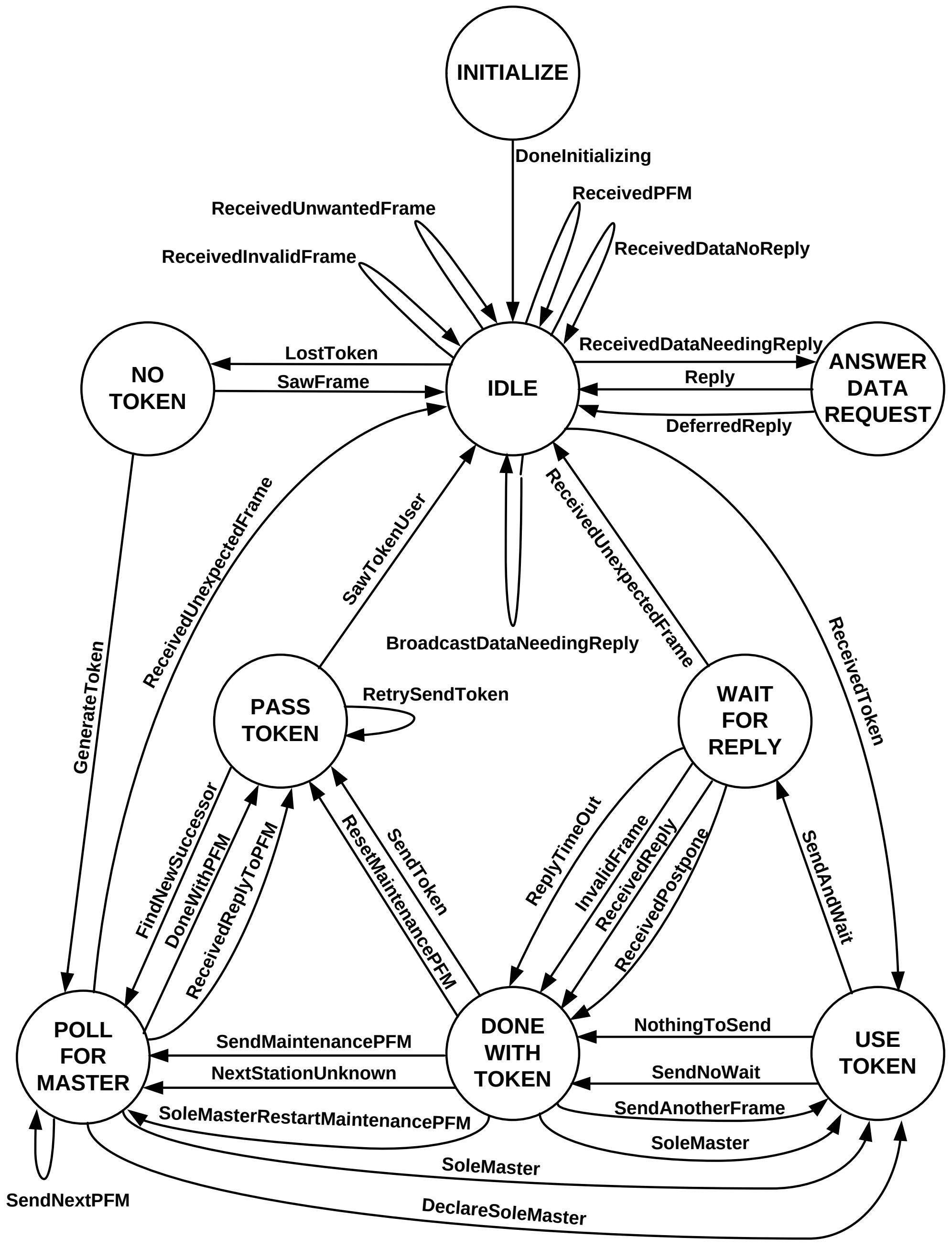
© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135-2012*bg*-9. Stop MS/TP nodes from sending POLL_FOR_MASTER frames to themselves.

Rationale

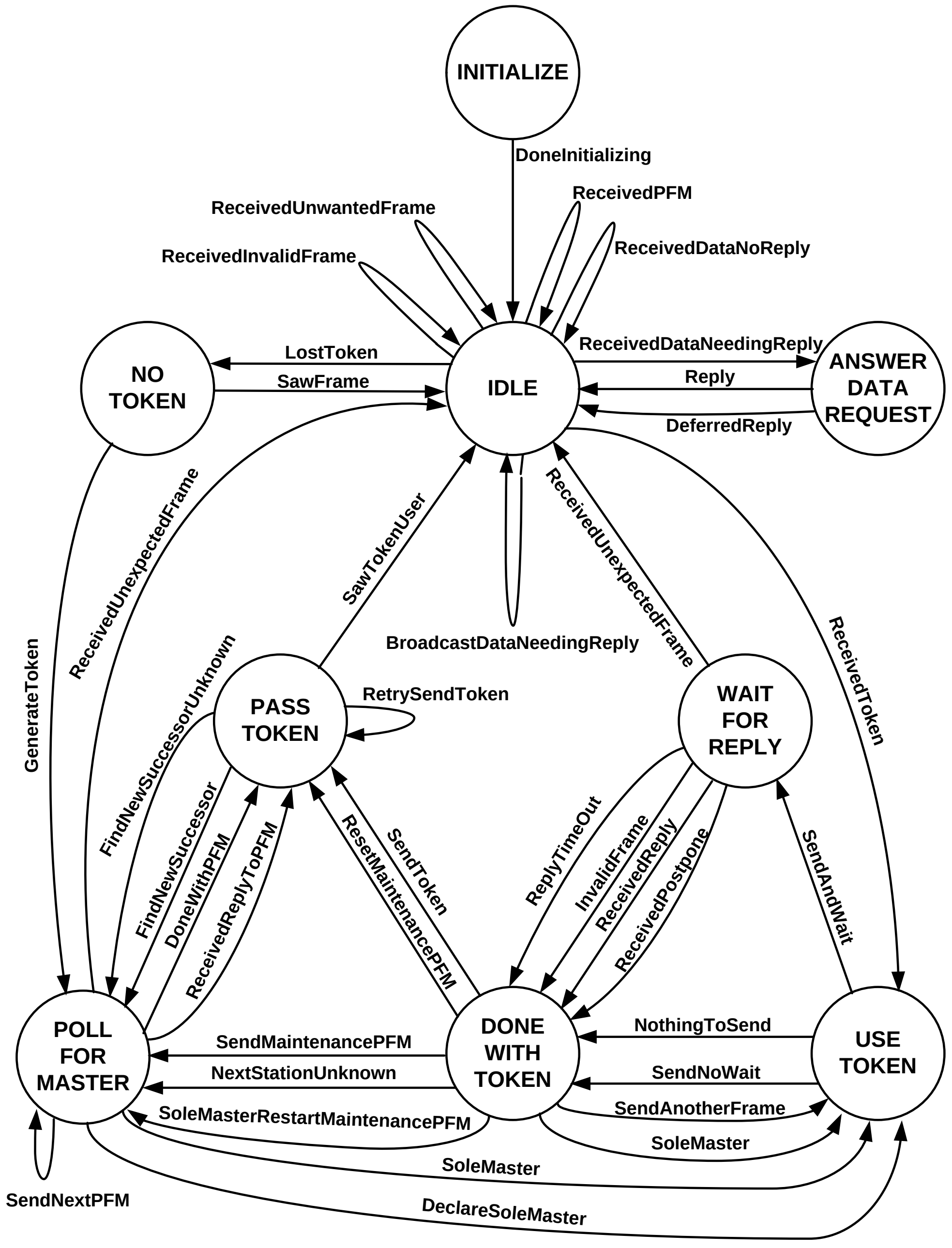
Under certain conditions, an MS/TP master node may send a Poll-For-Master frame addressed to itself. As a result, an invalid packet is emitted onto the network. When this occurs, there are no other nodes on the network, so the result has no consequence. This behavior was observed at a BTL test lab when removing the NextStation node when the NextStation node is at address ThisStation-1.

[Change **Figure 9-4**, p. 86]
[current figure]



© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

[revised figure]



[Change **Clause 9.5.6.6**, p. 107]

9.5.6.6 PASS_TOKEN

The PASS_TOKEN state listens for a successor to begin using the token that this node has just attempted to pass.

SawTokenUser

If SilenceTimer is less than $\mathbf{T}_{\mathbf{usage\_timeout}}$ and EventCount is greater than $\mathbf{N}_{\mathbf{min\_octets}}$,

then assume that a frame has been sent by the new token user. Enter the IDLE state to process the frame.

RetrySendToken

If SilenceTimer is greater than or equal to $\mathbf{T}_{\mathbf{usage\_timeout}}$ and RetryCount is less than $\mathbf{N}_{\mathbf{retry\_token}}$,

then increment RetryCount; call SendFrame to transmit a Token frame to NS; set EventCount to zero; and re-enter the current state to listen for NS to begin using the token.

FindNewSuccessorUnknown

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

If SilenceTimer is greater than or equal to $\boldsymbol{T}_{\boldsymbol{usage\_timeout}}$ *and RetryCount is greater than or equal to* $\boldsymbol{N}_{\boldsymbol{retry\_token}}$*, and (NS+1) modulo (*$\boldsymbol{N}_{\boldsymbol{max\_master}}$*+1) is equal to TS,*

then assume that NS has failed. Set PS to (TS+1) modulo ($\boldsymbol{N}_{\boldsymbol{max\_master}}$*+1); call SendFrame to transmit a Poll For Master frame to PS; set NS to TS (no known successor node); set RetryCount and TokenCount to zero; and enter the POLL_FOR_MASTER state to find a new successor to TS.*

FindNewSuccessor

If SilenceTimer is greater than or equal to $\mathbf{T}_{\mathbf{usage\_timeout}}$ and RetryCount is greater than or equal to $\mathbf{N}_{\mathbf{retry\_token}}$,

then assume that NS has failed. Set PS to (NS+1) modulo ($\mathbf{N}_{\mathbf{max\_master}}$+1); call SendFrame to transmit a Poll For Master frame to PS; set NS to TS (no known successor node); set RetryCount and TokenCount to zero; and enter the POLL_FOR_MASTER state to find a new successor to TS.

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135-2012*bg*-10. Improve the Clause 12 preamble.

Rationale

The preamble to Clause 12 is the only location where many key concepts of the BACnet object model are defined, but it has no subclause structuring to reference, so important information is easily missed.

These changes have no substantive impact and are entirely editorial in nature.

[Change **Clause 12**, p. 144]

12. MODELING CONTROL DEVICES AS A COLLECTION OF OBJECTS

The data structures used in a device to store information are a local matter. In order to exchange that information with another device using this protocol, there must be a "network-visible" representation of the information that is *accessible from a* standardized *mechanism*. An object-oriented approach has been adopted to provide this network-visible representation. This clause defines a set of standard object types. These object types define an abstract data structure that provides a framework for building the application layer services. The application layer services are designed, in part, to access and manipulate the properties of these standard object types~~.~~ *as well as the properties of nonstandard object types*. Mapping the effect of these services to the real data structures used in the device is a local matter. The number of instances of a particular object type that a device will support is also a local matter. *Objects provide and accept information through a collection of properties that may use various datatypes. It is intended that the collection of object types and their properties defined in this standard be comprehensive, but implementors are free to define additional nonstandard object types or additional nonstandard properties of standard object types. This is the principal means for extending the standard as control technology develops. Innovative changes can be accommodated without waiting for changes in the standard. This extensibility could also be used to adapt this standard to other types of building services. See Clauses 23.3 and 23.4.*

12.1 Object Characteristics and Requirements

All objects are subject to a common set of rules and requirements.

12.1.1 Identification of Objects

All objects are referenced by their Object_Identifier property *and every object shall have an Object_Identifier property. The Object_Identifier is composed of two parts: an object type and an object instance (see 20.2.14).* Each object within a single BACnet Device shall have a unique value for the Object_Identifier property. ~~When combined with the system-wide unique Object_Identifier of the BACnet Device, this provides a mechanism for referencing every object in the control system network.~~ No object shall have an Object_Identifier with an instance number of 4194303. Object properties that contain ~~BACnetObjectIdentifiers~~ *values whose datatype is BACnetObjectIdentifier* may use 4194303 *as the instance number* to indicate that the property is not initialized.

12.1.1.1 Object Type

Every object shall have an Object_Type property whose value shall be an enumeration equal to the object type portion of the Object_Identifier.

12.1.1.2 Object Name

Every object shall have an Object_Name property which is a character string. Object_Name properties shall be unique within the BACnet device that maintains them. The Object_Name string shall be at least one character in length and shall consist only of printable characters.

12.1.1.3 Device Instance

Each BACnet device shall have one and only one Device object instance. The object type of this object shall be DEVICE and the instance number shall be a internetwork-wide unique instance number. This is the Device Instance of the BACnet device. When an Object_Identifier is combined with the internetwork-wide unique Object_Identifier of the BACnet Device (having an object type DEVICE and the Device Instance), this provides a mechanism for referencing every object in the BACnet internetwork. Because of this requirement, every BACnet device shall provide a means of altering the Device Instance to assure uniqueness. The altered Device Instance shall be persistent across power failure and restart of the device.

12.1.1.4 Device Name

The Object_Name property of the Device object (the Device Name) shall be an internetwork-wide unique character string. Because of this requirement, every BACnet device shall provide a means of altering the Device Name to assure uniqueness. The altered Device Name shall be persistent across power failure and restart of the device.

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

12.1.1.5 Property_List

Every object shall have a Property_List property which is a BACnetARRAY of property identifiers, one property identifier for each property that exists within the object. The Object_Name, Object_Type, Object_Identifier, and Property_List properties are not included in the list.

12.1.2 Object Type and Property Conformance

Not all object types defined in this standard need to be supported in order to conform to the standard. *Other than the objects required by Clause 22.1.5, every BACnet device may choose any combination of standard and nonstandard object types and instances of those types that are appropriate to the device's application. In addition, object types may depend on a combination of particular properties. Some properties may be optionally implemented for particular objects as an implementor decision.* ~~In addition, some properties of particular object types are optional. At the beginning of each standard object type specification that follows is a summary of the properties of the object type. The summary includes the property identifier, the datatype of the property, and one of the following : **O**, **R**, **W**~~

~~where **O** indicates that the property is optional,~~
~~**R** indicates that the property is required to be present and readable using BACnet services,~~
~~**W** indicates that the property is required to be present, readable, and writable using BACnet services.~~

*Each standard object type includes a property table summarizing the standard properties that are relevant to that object type. The property table lists each property that is defined for that object. Each property includes the property identifier, the datatype of the property, and one of the following: **O**, **R**, **W**. A conformance code of **O** indicates that the property is optional; **R** indicates that the property is required to be present and readable using BACnet services; **W** indicates that the property is required to be present, readable, and writable using BACnet services.*

When a property is designated as required or **R**, this shall mean that the property is required to be present in all BACnet standard objects of that type. When a property is designated as optional or **O**, this shall mean that the property is not required to be present in all standard BACnet objects of that type. The value of **R** or **O** properties may be examined through the use of one or more of the ReadProperty services defined in this standard. Such **R** or **O** properties may also be writable at the implementor's option unless specifically prohibited in the text describing that particular standard object's property. When a property is designated as writable or **W**, this shall mean that the property is required to be present in all BACnet standard objects of that type and that the value of the property can be changed through the use of one or more of the WriteProperty services defined in this standard. The value of **W** properties may be examined through the use of one or more of the ReadProperty services defined in this standard. An **O** property, if present in a particular object, is not required to be writable unless specifically identified as such in the text describing that particular standard object's property.

~~In some devices, property values may be stored internally in a different form than indicated by the property datatype. For example, real numbers may be stored internally as integers. This may result in the situation where a property value is changed by one of the WriteProperty services but a subsequent read returns a slightly different value. This behavior is acceptable as long as a "best effort" is made to store the written value specified.~~

~~It is intended that the collection of object types and their properties defined in this standard be comprehensive, but implementors are free to define additional nonstandard object types or additional nonstandard properties of standard object types. This is the principal means for extending the standard as control technology develops. Innovative changes can be accommodated without waiting for changes in the standard. This extensibility could also be used to adapt this standard to other types of building services. See 23.3 and 23.4.~~

~~Nonstandard object types are required to support the following properties:~~

~~• Object_Identifier~~	~~BACnetObjectIdentifier~~
~~• Object_Name~~	~~CharacterString~~
~~• Object_Type~~	~~BACnetObjectType~~
~~• Property List~~	~~BACnetARRAY of BACnetPropertyIdentifier~~

~~These properties shall be implemented to behave as they would when present in standard BACnet objects. This means that the Object_Identifier and Object_Name properties shall be unique within the BACnet device that maintains them. The Object_Name string shall be at least one character in length and shall consist only of printable characters.~~

12.1.3 Required and Optional Properties

A BACnet standard object shall support all required properties specified in the ~~standard~~ *clause for its object type.* It may support, in addition to these properties, any optional properties specified in the ~~standard~~ *clause for its object type* or properties

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

not defined in the standard. A required property shall function as specified in the standard for each object of that type. If properties that are defined as optional in the standard are supported, then they shall function as specified in the standard. A required property *for a given object type* shall be present in all objects of that type. An optional property, if present in one object of a given type, need not be present in all objects of that type. A supported property, whether required or optional, shall return the datatype specified in the standard. A supported property, whether required or optional, is not required to be able to return the entire range of values for a datatype unless otherwise specified in the property description. Supported properties, whether required or optional, which do not return the entire range of values for a datatype when read or which restrict the range of values that may be written to the property, shall specify those restrictions for each such property in the protocol implementation conformance statement (PICS).

12.1.3.1 Required Properties Introduced in New Protocol Revisions

Required properties may change from revision to revision. Clients that depend on property requirements should take into account the Protocol_Revision, recognizing that a required property in some revision may not have existed in prior revisions.

12.1.4 Asymmetry in Property Values

In some devices, property values may be stored internally in a different form than indicated by the property datatype. For example, real numbers may be stored internally as integers, or with a more limited precision. This may result in the situation where a property value is changed by one of the WriteProperty services but a subsequent read returns a slightly different value. This behavior is acceptable as long as a "best effort" is made to store the written value specified.

12.1.5 Array and List Properties

Some of the properties of certain BACnet objects need to represent a collection of data elements of the same type, rather than a single primitive data value or a complex datatype constructed from other datatypes. In some instances, the size of this collection of data elements is fixed, while in other instances the number of elements may be variable. In some cases the elements may need to be accessed individually or their order may be important. BACnet provides two forms of datatypes for properties that represent a collection of data elements of the same type: "BACnetARRAY" and "BACnetLIST." ~~Both "BACnetARRAY" and "List of" are encoded as a "Sequence-Of". Therefore, see the note about datatype restrictions in Clause 20.2.17.~~

12.1.5.1 Array Properties

A "BACnetARRAY" datatype is a structured datatype consisting of an ordered sequence of data elements, each having the same datatype. The components of an array property may be individually accessed (read or written) using an "array index," which is an unsigned integer value. An index of 0 (zero) shall specify ~~that~~ the count of the number of data elements ~~be returned~~ *as datatype Unsigned*. If the array index is omitted, it means that all of the elements of the array are to be accessed. An array index N, greater than zero, shall specify the Nth element in the sequence. When array properties are used in BACnet objects, the notation "BACnetARRAY[N] of datatype" shall mean an ordered sequence of N data elements, *beginning with index 1,* each of which has that datatype. ~~The datatype of array element 0 is Unsigned~~.
If the size of an array may be changed by writing to the array, then array element 0 shall be writable. If the value of array element 0 is decreased, *for those arrays whose size is writable,* the array shall be truncated and the elements of the array with an index greater than the new value of array element 0 are deleted. If the value of array element 0 is increased, *for those arrays whose size is writable*, the new elements of the array, ~~those~~ with an index greater than the old value of array element 0, shall be created; the values that are assigned to those elements shall be a local matter except where otherwise specified. Where the size of an array is allowed to be changed, writing the entire array as a single property with a different number of elements *than the existing array* shall cause the array size to be changed. An attempt to write to an array element with an index greater than the size of the array shall result in an error and shall not cause the array to grow to accommodate the element. Arrays whose sizes are fixed by the Standard shall not be resizable.

12.1.5.2 List Properties

A "BACnetLIST" datatype is a structured datatype consisting of a sequence of zero or more data elements, each having the same datatype. The length of each "BACnetLIST" may be variable. Unless specified for a particular use, no maximum size should be assumed for any “BACnetLIST" implementation. The notation "BACnetLIST of datatype" shall mean a sequence of zero or more data elements, each of which has the indicated type**.**

The difference between a "BACnetARRAY" property and a "BACnetLIST" property is that the elements of the array can be uniquely accessed by an array index while the elements of the "BACnetLIST" property can only be positionally accessed using the ReadRange service. Moreover, the number of elements in the BACnetARRAY may be ascertained by reading the array index 0, while the number of elements present in a "BACnetLIST" property can only be determined by reading the entire property value and performing a count.

~~The ordering of list elements when a list is written or modified is not required to be preserved upon subsequent reading of the same list, even if the set of elements that make up the list has not changed.~~

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution, or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

In the context of ReadRange 'By Position', the ordering of BACnetLIST elements shall follow the conventions that the first element of the BACnetLIST shall be position 1, and positions 2, 3, 4 and greater shall correspond to list elements in strict sequence. The sequence of list elements shall follow the same ordering that those elements would appear in if the entire list was read using ReadProperty to read the entire list. *The order of list elements when a list is written or modified is not required to be preserved by the device. However,* ~~Assuming~~ *assuming* that the list has not been written or modified, repeated reading of list elements shall return those elements in the same order each time.

When adding items to a "BACnetLIST" via AddListElement, only unique items shall be added. For example a list {"A", "B"} to which "A" is added will remain the same. However, particular properties may specify special criteria whereby only a portion of each list element is compared for uniqueness when checking for equality.

12.1.6 Special Property Identifiers

Several object types defined in this clause - *for example, the Command, Event Enrollment, Group, Loop, and Schedule object types* - have one or more properties that are capable of referencing object properties. ~~For example, the Object_Property_Reference property of the Event Enrollment object contains such a reference.~~ The property identifier component of these references shall not be ~~any of~~ the special property identifiers ALL, REQUIRED, or OPTIONAL. These are reserved for use in the ReadPropertyMultiple service or in ~~objects and~~ services not defined in this standard.

12.1.7 Unspecified Dates and Times

Several object types defined in this clause have properties that are of type BACnetDateTime, and specify a specific point in time. These properties shall have an unspecified datetime value if the point in time is undefined or a specific datetime value if the point in time is specified.

There are a number of objects defined in this clause that have properties of the type BACnetDateRange, for example, the Date_List property in the Calendar object, whose construct includes a startDate and an endDate. Both startDate and endDate may be an unspecified date or a specific date only. For purposes of comparing date ranges, the following logic shall be applied.

The use of an unspecified date in the startDate means "any date up to and including the endDate." The use of an unspecified date in the endDate means "any date after and including the startDate." The use of an unspecified date in both the startDate and the endDate means "any date" or "always."

Several object types defined in this clause have properties that contain timestamp values. If no event or operation has yet occurred, then timestamp values of type BACnetDateTime shall have an unspecified datetime value, timestamp values of type Time shall have an unspecified time value, and timestamp values of type Unsigned shall have a value of zero. If the event or operation has occurred, then the timestamp value shall have a specific datetime value, a specific time value, or a value greater than zero, respectively. If a device supports the Local_Date and Local_Time properties, then all timestamps created by the device shall use the BACnetDateTime form.

12.1.8 Reliability

Several object types defined in this clause have a property called "Reliability" that indicates the existence of fault conditions for the object. Reliability-evaluation is the process of determining the value for this property. The first stage of reliability-evaluation is internal to the object and is completely defined by the device's vendor. The second stage, which is only found in certain object types, is the application of a fault algorithm. See Clause 13.4 for fault algorithm definitions and see the object type definitions to determine the algorithm supported by any particular object type. The different values that the Reliability property can take on are described below. Note that not all values are applicable to all object types.

NO_FAULT_DETECTED	The present value is reliable; that is, no other fault (enumerated below) has been detected.
NO_SENSOR	No sensor is connected to the Input object.
OVER_RANGE	The sensor connected to the Input is reading a value higher than the normal operating range. If the object is a Binary Input, this is possible when the Binary state is derived from an analog sensor or a binary input equipped with electrical loop supervision circuits.
UNDER_RANGE	The sensor connected to the Input is reading a value lower than the normal operating range. If the object is a Binary Input, this is possible when the Binary Input is actually a binary state calculated from an analog sensor.

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution,
or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

OPEN_LOOP	The connection between the defined object and the physical device is providing a value indicating an open circuit condition.
SHORTED_LOOP	The connection between the defined object and the physical device is providing a value indicating a short circuit condition.
NO_OUTPUT	No physical device is connected to the Output object.
PROCESS_ERROR	A processing error was encountered.
MULTI_STATE_FAULT	The FAULT_STATE, FAULT_LIFE_SAFETY or FAULT_CHARACTERSTRING fault algorithm has evaluated a fault condition. For details of this evaluation see the respective fault algorithms in Clause 13.4.
CONFIGURATION_ERROR	The object's properties are not in a consistent state.
COMMUNICATION_FAILURE	Proper operation of the object is dependent on communication with a remote sensor or device and communication with the remote sensor or device has been lost.
MONITORED_OBJECT_FAULT	Indicates that the monitored object is in fault.
UNRELIABLE_OTHER	The controller has detected that the present value is unreliable, but none of the other conditions describe the nature of the problem. A generic fault other than those listed above has been detected, e.g., a Binary Input is not cycling as expected.
MEMBER_FAULT	Indicates that the set of referenced member objects includes one or more Status_Flags properties whose FAULT flag value is equal to TRUE.
TRIPPED	The end device, such as an actuator, is not responding to commands, prevented by a tripped condition or by being mechanically held open.

[Move **Clause 12.1** to **12.X**, p. 459]
[Note: This change is made so as to minimize the renumbering of object type definition clauses.]

~~12.1~~*12.X* Accumulator Object Type

The Accumulator object type defines a standardized object whose properties represent the externally visible characteristics of a device that indicates measurements made by counting pulses

...

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution,
or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

135-2012*bg*-11. Fix the Notification_Class property of the Notification Class object.

Rationale

Addendum section 135-2012*aw*-6 adds fault reporting to the Notification Class object. It proposes new language for the Notification_Class property, consistent with the general language used for this property in other object types which report events. The fact that the Notification Class object already has a Notification_Class property whose purpose is different was missed.

The current definition of the Notification Class's Notification_Class property just reflects the instance number of the Notification Class object. In contrast, fault reporting needs a Notification_Class property which indicates the Notification Class object through which events are to be reported. In most cases, it does not make sense that the Notification Class object use its own Recipient_List for reporting faults as it is expected that most faults will be due to the Notification Class object's inability to report events to its recipients.

Therefore, It is proposed that the Notification Class object's Notification_Class property be changed to behave in the same manner in which it behaves in all event generating objects. This is accomplished by using the general definition and language for the Notification_Class property as of Addendum 135-2012aw-6.

[Change **Clause 12.21.5**, p. 258]
[Note to reviewer: The proposed new language for the Notification Class object's Notification_Class property was already published with Addendum 135-2012aw-6 for the Notification Class object.]

12.21.5 Notification_Class

~~This property, of type Unsigned, shall indicate the numeric value of this notification class and shall be equal to the instance number of the Notification Class object. Event-initiating objects shall use this number to refer to this Notification Class object.~~

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

[Change **Clause 13.11.1.2.1.5**, p. 523]

13.11.1.2.1.5 Notification Class

This optional parameter, of type Unsigned, indicates the class of notifications generated by the object and ~~implicitly~~ refers to a Notification Class object that has ~~a Notification_Class property~~ *an object instance number* of the same value.

© ASHRAE (www.ashrae.org). For personal use only. Additional reproduction, distribution,
or transmission in either print or digital form is not permitted without ASHRAE's prior written permission.

[Add a new entry to **History of Revisions**, p. 1027]

(This History of Revisions is not part of this standard. It is merely informative and does not contain requirements necessary for conformance to the standard.)

HISTORY OF REVISIONS

…	…	**…**
1	18	**Addendum *bg* to ANSI/ASHRAE 135-2012** Approved by ASHRAE on February 29, 2016, and by the American National Standards Institute March 1, 2016. 1. Add engineering units 2. Harmonize the message text handling for all alarm services 3. Ensure Alert Enrollment objects do not send notifications which require acknowledgment 4. Allow selection of the Nth last day of the month in a BACnetWeekNDay 5. Remove initiation of GetEnrollmentSummary from AE-AS-A 6. Ensure UTC_Offset is configurable 7. Clarify ReadRange 8. Clarify the effect of changing Buffer_Size 9. Stop MS/TP nodes from sending POLL_FOR_MASTER frames to themselves 10. Improve the Clause 12 preamble 11. Fix the Notification_Class property of the Notification Class object

POLICY STATEMENT DEFINING ASHRAE'S CONCERN FOR THE ENVIRONMENTAL IMPACT OF ITS ACTIVITIES

ASHRAE is concerned with the impact of its members' activities on both the indoor and outdoor environment. ASHRAE's members will strive to minimize any possible deleterious effect on the indoor and outdoor environment of the systems and components in their responsibility while maximizing the beneficial effects these systems provide, consistent with accepted Standards and the practical state of the art.

ASHRAE's short-range goal is to ensure that the systems and components within its scope do not impact the indoor and outdoor environment to a greater extent than specified by the Standards and Guidelines as established by itself and other responsible bodies.

As an ongoing goal, ASHRAE will, through its Standards Committee and extensive Technical Committee structure, continue to generate up-to-date Standards and Guidelines where appropriate and adopt, recommend, and promote those new and revised Standards developed by other responsible organizations.

Through its *Handbook*, appropriate chapters will contain up-to-date Standards and design considerations as the material is systematically revised.

ASHRAE will take the lead with respect to dissemination of environmental information of its primary interest and will seek out and disseminate information from other responsible organizations that is pertinent, as guides to updating Standards and Guidelines.

The effects of the design and selection of equipment and systems will be considered within the scope of the system's intended use and expected misuse. The disposal of hazardous materials, if any, will also be considered.

ASHRAE's primary concern for environmental impact will be at the site where equipment within ASHRAE's scope operates. However, energy source selection and the possible environmental impact due to the energy source and energy transportation will be considered where possible. Recommendations concerning energy source selection should be made by its members.

ASHRAE · 1791 Tullie Circle NE · Atlanta, GA 30329 · www.ashrae.org

About ASHRAE

ASHRAE, founded in 1894, is a global society advancing human well-being through sustainable technology for the built environment. The Society and its members focus on building systems, energy efficiency, indoor air quality, refrigeration, and sustainability. Through research, Standards writing, publishing, certification and continuing education, ASHRAE shapes tomorrow's built environment today.

For more information or to become a member of ASHRAE, visit www.ashrae.org.

To stay current with this and other ASHRAE Standards and Guidelines, visit www.ashrae.org/standards.

Visit the ASHRAE Bookstore

ASHRAE offers its Standards and Guidelines in print, as immediately downloadable PDFs, on CD-ROM, and via ASHRAE Digital Collections, which provides online access with automatic updates as well as historical versions of publications. Selected Standards and Guidelines are also offered in redline versions that indicate the changes made between the active Standard or Guideline and its previous version. For more information, visit the Standards and Guidelines section of the ASHRAE Bookstore at www.ashrae.org/bookstore.

IMPORTANT NOTICES ABOUT THIS STANDARD

To ensure that you have all of the approved addenda, errata, and interpretations for this Standard, visit www.ashrae.org/standards to download them free of charge.

Addenda, errata, and interpretations for ASHRAE Standards and Guidelines are no longer distributed with copies of the Standards and Guidelines. ASHRAE provides these addenda, errata, and interpretations only in electronic form to promote more sustainable use of resources.

3/16