ANSI/ASHRAE Addendum u
to ANSI/ASHRAE Standard 135-2008

ASHRAE STANDARD

BACnet®—A Data Communication Protocol for Building Automation and Control Networks

Approved by the ASHRAE Standards Committee on January 23, 2010; by the ASHRAE Board of Directors on January 27, 2010; and by the American National Standards Institute on January 28, 2010.

This standard is under continuous maintenance 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 Web site, http://www.ashrae.org, or in paper form from the Manager of Standards. The latest edition of an ASHRAE Standard may be purchased from ASHRAE Customer Service, 1791 Tullie Circle, NE, Atlanta, GA 30329-2305. E-mail: orders@ashrae.org. Fax: 404-321-5478. Telephone: 404-636-8400 (worldwide), or toll free 1-800-527-4723 (for orders in US and Canada).

ISSN 1041-2336

Approved American National Standard
ANSI
www.ansi.org

American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.
1791 Tullie Circle NE, Atlanta, GA 30329
www.ashrae.org

ASHRAE Standing Standard Project Committee 135
Cognizant TC: TC 1.4, Control Theory and Application
SPLS Liaison: Douglas T. Reindl

David Robin, *Chair**	Troy Cowan	Jerald P. Martocci
Carl Neilson, *Vice-Chair*	Harsha M. Dabholkar	Cherisse M. Nicastro
Sharon E. Dinges, *Secretary**	Thomas Ertsgaard	Mark A. Railsback
Donald P. Alexander*	Craig P. Gemmill	Carl J. Ruther
Beauford W. Atwater	Daniel P. Giorgis	Patrick F. Sheridan
David J. Branson	Ira G. Goldschmidt	David G. Shike
Barry B. Bridges*	David G. Holmberg	Ted Sunderland
Coleman L. Brumley, Jr.*	Bernhard Isler*	William O. Swan, III
Ernest C. Bryant	Robert L. Johnson	David B. Thompson*
Steven T. Bushby	Stephen Karg*	Daniel A. Traill
James F. Butler	Simon Lemaire	Stephen J. Treado*
A. J. Capowski	J. Damian Ljungquist*	J. Michael Whitcomb*
Keith A. Corbett	James G. Luth	David F. White
Jeffrey Cosiol	John J. Lynch	Grant N. Wichenko

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

SPECIAL NOTE

This American National Standard (ANS) is a national voluntary consensus standard developed under the auspices of the American Society of Heating, Refrigerating and Air-Conditioning Engineers (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 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.

[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

Addendum 135*u* to ANSI/ASHRAE Standard 135-2008 contains a number of changes to the current standard. 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 changes are summarized below.

In the following document, language added to existing clauses of ANSI/ASHRAE 135-2008 and addenda is indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are added, plain type is used throughout.

135-2008*u*-1. Clarify the use of RejectPDUs.

Rationale
There has been some confusion over the use of RejectPDUs. Some interpreted the standard to mean that a RejectPDU is only issued when a service cannot be executed at all, and as such, a service that is rejected would have no impact on a device (other than possibly logging/statistics). Others interpreted the standard to mean that a RejectPDU could be sent at any point in the execution of a service if a syntax error is encountered. A specific case where this confusion could arise is with a malformed WritePropertyMultiple service request. A device interpreting the standard in the latter way could return a Reject PDU after it had applied some of the write request. WritePropertyMultiple requests, and possibly AtomicWriteFile requests, are the only services where the service may be partially executed such that a partial execution changes the state of the device. For that reason these are the only two services that are impacted by this change proposal.

Addendum 135-2008*u*-1

[Change Clause **5.4.4.2**, p. 28]

...

RejectPDU_Received
If a BACnet-Reject-PDU is received from the network ~~layer and SentAllSegments is TRUE,~~ *layer,*

then stop SegmentTimer; send REJECT.indication to the local application program; and enter the IDLE state.

UnexpectedPDU_Received
If a BACnet-SimpleACK-PDU, BACnet-ComplexACK-PDU, *or BACnet-Error-PDU* ~~BACnet-Error-PDU, or BACnet-Reject-PDU~~ is received from the network layer and SentAllSegments is FALSE,

or if a BACnet-ComplexACK-PDU is received from the network layer whose 'segmented-message' parameter is TRUE and this device does not support segmentation,

or if a BACnet-ComplexACK-PDU is received from the network layer whose 'segmented-message' parameter is TRUE and whose 'sequence-number' parameter is not zero,

then stop SegmentTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = FALSE; send CONF_SERV.confirm(-) to the local application program; and enter the IDLE state.

...

[Add to the table in Clause **14.2.4.1**. p. 335]

Situation:	Error Class:	Error Code:
…		
A syntax error is encountered in the message after the file has been partially modified during the execution of this service.	*SERVICES*	*INVALID_TAG*

[Add to the table in Clause **15.10.1.3.1**. p. 358]

Situation	Error Class	Error Code
…		
A syntax error is encountered in the message after one or more properties have been successfully written.	*SERVICES*	*INVALID_TAG*

[Change Clause **20.1.7**, p. 413]

20.1.7 BACnet-Error-PDU

The BACnet-Error-PDU is used to convey the information contained in a service response primitive ('Result(-)') that indicates the reason why a previous confirmed service request failed *either* in its entirety *or only partially*.

…

[Change Clause **20.1.8**, p. 414]

20.1.8 BACnet-Reject-PDU

The BACnet-Reject-PDU is used to reject a received confirmed request PDU based on syntactical flaws or other protocol errors that prevent the PDU from being interpreted or the requested service from being provided. Only confirmed request PDUs may be rejected (see 18.8). *A BACnet-Reject-PDU shall be sent only before the execution of the service, such as during the interval after a syntax check is performed on the request but before the service procedure is executed. Such a syntax check may occur as segments are received and thus may result in a BACnet-Reject-PDU being returned before the complete request has been received.*

[Change Clause **18.6**, p. 393]

18.6 Error Class - SERVICES

This Error Class pertains to problems related to the execution of protocol service requests, whether BACnet-defined or not. ~~Without exception, these errors signal the inability of the responding BACnet-user to carry out the desired service in its entirety and are thus "fatal."~~

[Add new Clause **18.6.X**, p. 394]

18.6.X INVALID TAG

This error indicates that a syntax error was encountered in the request.

135-2008*u*-2. Add error code UNSUPPORTED_OBJECT_TYPE for CreateObject service.

Rationale
The CreateObject service error codes do not include the error code UNSUPPORTED_OBJECT. Instead, the table requires returning DYNAMIC_CREATION_NOT_SUPPORTED, which is not as accurate in reporting the error condition.

Addendum 135-2008*u*-2

[Change Clause **15.3.1.3.1**, p. 341]

Situation	Error Class	Error Code
The device cannot allocate the space needed for the new object.	RESOURCES	NO_SPACE_FOR_OBJECT
~~The device does not support creation of this object for any reason other than space~~. *The device supports the object type and may have sufficient space, but does not support the creation of the object for some other reason.*	OBJECT	DYNAMIC_CREATION_NOT_SUPPORTED
The device does not support the specified object type.	*OBJECT*	*UNSUPPORTED_OBJECT_TYPE*
The object being created already exists.	OBJECT	OBJECT_IDENTIFIER_ALREADY_EXISTS
A datatype of a property value specified in the List of Initial Values does not match the datatype of the property specified by the Property_Identifier.	PROPERTY	INVALID_DATATYPE
A value used in the List of Initial Values is outside the range of values defined for the property specified by the Property_Identifier.	PROPERTY	VALUE_OUT_OF_RANGE
A Property_Identifier has been specified in the List of Initial Values that is unknown for objects of the type being created.	PROPERTY	UNKNOWN_PROPERTY
A character string value was encountered in the List of Initial Values that is not a supported character set.	PROPERTY	CHARACTER_SET_NOT_SUPPORTED
A property specified by the Property_Identifier in the List of Initial Values does not support initialization during the CreateObject service.	PROPERTY	WRITE_ACCESS_DENIED
The data being written has a datatype not supported by the property.	PROPERTY	DATATYPE_NOT_SUPPORTED

135-2008*u*-3 Add new Abort and Error codes.

Rationale
There are various reasons for services and messages to return failure indications. The present standard defines a set of ABORT codes and ERROR codes. The problem is that there are many scenarios that have been encountered in practice for which no existing code is suitable, or is at best fairly vague. The existing error enumerations do not address a number of common scenarios, thereby forcing implementers to use vague error returns like Error-Other. The inclusion of additional error enumerations will be extremely helpful in troubleshooting and diagnosing these kinds of errors by adding specificity to error indication.

Addendum 135-2008*u*-3

[Change Clause **5.4.4.1**, p. 26]

…

CannotSend

If CONF_SERV.request is received from the local application program and the length of the APDU is greater than maximum-transmittable-length as determined according to 5.2.1 and the Max_Segments_Accepted property of the destination's Device object is known and the total APDU cannot be transmitted without exceeding the maximum number of segments accepted,

then send an ABORT.indication *with 'server' = FALSE and 'abort-reason' =APDU_TOO_LONG* to the local application program and enter the IDLE state.

[Change Clause **5.4.4.2**, p. 27]

…

FinalTimeout

If SegmentTimer becomes greater than T_{seg} and SegmentRetryCount is greater than or equal to N_{retry},

then stop SegmentTimer; send ~~CONF_SERV.confirm( )~~ *ABORT.indication with 'server' = FALSE and 'abort-reason' = TSM_TIMEOUT* to the local application program; and enter the IDLE state.

...

[Change Clause **5.4.4.2**, p. 28]

…

UnexpectedPDU_Received

If a BACnet-SimpleACK-PDU, BACnet-ComplexACK-PDU, BACnet-Error-PDU, or BACnet-Reject-PDU is received from the network layer and SentAllSegments is FALSE,
or if a BACnet-ComplexACK-PDU is received from the network layer whose 'segmented-message' parameter is TRUE and this device does not support segmentation,

or if a BACnet-ComplexACK-PDU is received from the network layer whose 'segmented-message' parameter is TRUE and whose 'sequence-number' parameter is not zero,

then stop SegmentTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = FALSE; send ~~CONF_SERV.confirm( )~~ *ABORT.indication with 'server' = FALSE and 'abort-reason' = INVALID_APDU_IN_THIS_STATE* to the local application program; and enter the IDLE state.

...

[Change Clause **5.4.4.3**, p. 29]

…

UnexpectedPDU_Received

If an unexpected PDU (BACnet-ComplexACK-PDU with 'segmented-message' = TRUE and 'sequence-number' not equal to zero or 'segmented-message' = TRUE and this device does not support segmentation) is received from the network layer,

then stop RequestTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = FALSE; send ~~CONF_SERV.confirm( )~~ *ABORT.indication with 'server' = FALSE and 'abort-reason' = INVALID_APDU_IN_THIS_STATE* to the local application program; and enter the IDLE state.

...

[Change Clause **5.4.4.3**, p.30]

…

FinalTimeout

If RequestTimer becomes greater than T_{out} and RetryCount is greater than or equal to Number_Of_APDU_Retries,

then stop RequestTimer; send ~~CONF_SERV.confirm( )~~ *ABORT.indication with 'server' = FALSE and 'abort-reason' = TSM_TIMEOUT* to the local application program; and enter the IDLE state.

...

...

[Change Clause **5.4.4.4**, p. 30]

…

NewSegmentReceived_NoSpace

If a BACnet-ComplexACK-PDU is received from the network layer whose 'segmented-message' parameter is TRUE; whose 'sequence-number' parameter is equal to LastSequenceNumber plus 1, modulo 256; and the segment cannot be saved due to local conditions,

then stop SegmentTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = FALSE *and 'abort-reason' = BUFFER_OVERFLOW*; send ~~CONF_SERV.confirm( )~~ *ABORT.indication with 'server' = FALSE and 'abort-reason' = BUFFER_OVERFLOW* to the local application program; and enter the IDLE state.

…

[Change Clause **5.4.4.4**, p. 31]

…

UnexpectedPDU_Received

If an unexpected PDU (BACnet-SimpleACK-PDU, BACnet-ComplexACK-PDU with 'segmented-message' = FALSE, BACnet-Error-PDU, BACnet-Reject-PDU, or BACnet-SegmentACK-PDU with 'server' = TRUE) is received from the network layer,

then stop SegmentTimer; issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = FALSE; send ~~CONF_SERV.confirm( )~~ *ABORT.indication with 'server' = FALSE and 'abort-reason' = INVALID_APDU_IN_THIS_STATE* to the local application program; and enter the IDLE state.

Timeout

If SegmentTimer becomes greater than T_{seg} times four,

then stop SegmentTimer; send ~~CONF_SERV.confirm( )~~ *ABORT.indication with 'server' = FALSE and 'abort-reason' = TSM_TIMEOUT* to the local application program; and enter the IDLE state.

...

[Change Clause **5.4.5.1**, p. 32]

…

ConfirmedSegmentedReceived

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

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

ConfirmedSegmentedReceivedWindowSizeOutOfRange

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

then issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = TRUE and 'abort-reason' = WINDOW_SIZE_OUT_OF_RANGE, and enter the IDLE state.

…

[Change Clause **5.4.5.3**, p. 35]

…

UnexpectedPDU_Received

If an unexpected PDU (BACnet-SegmentACK-PDU whose 'server' parameter is FALSE) is received from the network layer,

then issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = TRUE; send ABORT.indication *with 'server' = TRUE and 'abort-reason' = INVALID_APDU_IN_THIS_STATE* to the local application program; and enter the IDLE state.

Timeout

If RequestTimer becomes greater than $\mathbf{T_{out}}$,

then issue an N-UNITDATA.request with 'data_expecting_reply' = FALSE to transmit a BACnet-Abort-PDU with 'server' = TRUE *and 'abort-reason' = APPLICATION_EXCEEDED_REPLY_TIME*; send ABORT.indication *with 'server' = TRUE and 'abort-reason' = APPLICATION_EXCEEDED_REPLY_TIME* to the local application program; and enter the IDLE state.

[Add entry to table in Clause **14.1.4.1**, p.332]

14.1.4.1 Error Type

This parameter consists of two component parameters: (1) the 'Error Class' and (2) the 'Error Code'. See Clause 18. The 'Error Class' and 'Error Code' to be returned for specific situations are as follows:

Situation:	Error Class:	Error Code:
The File object does not exist.	OBJECT	UNKNOWN_OBJECT
'File Start Record' is out of range.	SERVICES	INVALID_FILE_START_POSITION
Incorrect File access method.	SERVICES	INVALID_FILE_ACCESS_METHOD
A non-File Object Identifier was provided.	*SERVICES*	*INCONSISTENT_OBJECT_TYPE*

[Change Clause **14.2.4.1**, p. 335]

14.2.4.1 Error Type

This parameter consists of two component parameters: (1) the 'Error Class' and (2) the 'Error Code'. See Clause 18. The 'Error Class' and 'Error Code' to be returned for specific situations are as follows:

Situation:	Error Class:	Error Code:
The File object does not exist.	OBJECT	UNKNOWN_OBJECT
'File Start Record' is out of range.	SERVICES	INVALID_FILE_START_POSITION
Incorrect File access method.	SERVICES	INVALID_FILE_ACCESS_METHOD
Write to a read-only File.	SERVICES	FILE_ACCESS_DENIED
The File object is full	*OBJECT*	*FILE_FULL*
A non-File Object Identifier was provided	*SERVICES*	*INCONSISTENT_OBJECT_TYPE*

[Change Clause **15.8.1.3.1**, p. 355]

15.8.1.3.1 Error Type

This parameter consists of two component parameters: (1) the 'Error Class' and (2) the 'Error Code'. See Clause 18. *The 'Error Class' and 'Error Code' to be returned for specific situations are as follows:*

Situation	*Error Class*	*Error Code*
Specified property does not exist.	*PROPERTY*	*UNKNOWN_PROPERTY*
The specified property is currently not readable by the requester.	*PROPERTY*	*READ_ACCESS_DENIED*
Property is not a list or array of lists	*SERVICES*	*PROPERTY_IS_NOT_A_LIST*
An array index is provided but the property is not an array.	*PROPERTY*	*PROPERTY_IS_NOT_AN_ARRAY*
An array index is provided that is outside the range existing in the property.	*PROPERTY*	*INVALID_ARRAY_INDEX*

[Add new error codes to Clause **18.1**, p. 391]

18.1 Error Class - DEVICE

This Error Class pertains to circumstances that affect the functioning of an entire BACnet device. The presence of one of these errors generally indicates that the entire service request has failed.

…

18.1.W INTERNAL_ERROR - *There are cases where some internal error is encountered. These are cases that are never expected to occur, but if they do the manufacturer should be contacted.*

18.1.X NOT_CONFIGURED - *A device may require configuration, possibly vendor-specific, before it becomes functional. If it is not configured, it can receive BACnet requests but cannot reasonably process them.*

18.1.Y INCONSISTENT_CONFIGURATION - *This error code is used when a device is misconfigured and hence cannot process a request.*

[Add new error codes to Clause **18.2**, p. 391]

18.2 Error Class - OBJECT

This Error Class pertains to problems related to identifying, accessing, and manipulating BACnet objects, whether BACnet-defined or not. Since these errors generally apply to individual object characteristics, they do not necessarily signal that an entire service request has failed.

…

18.2.X FILE_FULL - *This applies to the case when a File Object becomes filled to a designed limit, as opposed to a No Space Available / Out of Memory situation.*

[Add new error codes to Clause **18.4**, p. 392]

18.4 Error Class - RESOURCES

This Error Class pertains to problems related to the resources of a BACnet device that affect its capacity to carry out protocol service requests.

…

18.4.X OUT_OF_MEMORY - *There are many internal operations during the processing of typical messages that may rely on acquiring dynamically allocated space. This indicates the failure of such an allocation.*

[Add new error codes to Clause **18.6**, p. 393]

18.6 Error Class - SERVICES

This Error Class pertains to problems related to the execution of protocol service requests, whether BACnet-defined or not. Without exception, these errors signal the inability of the responding BACnet-user to carry out the desired service in its entirety and are thus "fatal."

…

18.6.X INCONSISTENT_OBJECT_TYPE - *A device receives a service request for an object whose type is inconsistent with the service requested, or for an object that doesn't support the service. An example is an AtomicReadFile request received for an object that is not a File object.*

[Add new error codes to Clause **18.7**, p. 394]

18.7 Error Class - COMMUNICATIONS

This Error Class pertains to problems related to network communications. These codes indicate problems reported by a remote device in abort and reject PDUs, or they indicate problems detected internally. These error codes are stored in properties of objects whose operation involves the network communications, such as the Trend Log object's Log_Buffer property. This Error Class shall not be conveyed in error PDUs.

…

***18.7.V ABORT_WINDOW_SIZE_OUT_OF_RANGE** - A device receives a request that is segmented, or receives any segment of a segmented request, where the Proposed Window Size field of the PDU header is either zero or greater than 127.*

***18.7.W ABORT_APPLICATION_EXCEEDED_REPLY_TIME** - A device receives a confirmed request but its application layer has not responded within the published APDU Timeout period.*

***18.7.X ABORT_OUT_OF_RESOURCES** - A device receives a request but cannot start processing because it has run out of some internal resource.*

***18.7.Y ABORT_TSM_TIMEOUT** - A transaction state machine timer exceeded the timeout applicable for the current state, causing the transaction machine to abort the transaction.*

***18.7.Z ABORT_APDU_TOO_LONG** - An APDU was received from the local application program whose overall size exceeds the maximum transmittable length or exceeds the maximum number of segments accepted by the server.*

[Add new abort reasons to Clause **18.10**, p. 396]

18.10 Abort Reason

…

***18.10.V WINDOW_SIZE_OUT_OF_RANGE** - A device receives a request that is segmented, or receives any segment of a segmented request, where the Proposed Window Size field of the PDU header is either zero or greater than 127.*

***18.10.W APPLICATION_EXCEEDED_REPLY_TIME** - A device receives a confirmed request but its application layer has not responded within the published APDU Timeout period.*

***18.10.X OUT_OF_RESOURCES** - A device receives a request but cannot start processing because it has run out of some internal resource.*

***18.10.Y TSM_TIMEOUT** - A transaction state machine timer exceeded the timeout applicable for the current state, causing the transaction machine to abort the transaction.*

***18.7.Z APDU_TOO_LONG** - An APDU was received from the local application program whose overall size exceeds the maximum transmittable length or exceeds the maximum number of segments accepted by the server.*

[Add new Clause **18.X**, p. 397]

18.X Confirmed Service Common Errors

Some errors are generic and can occur when any confirmed service is requested. The 'Error Class' and 'Error Code' to be returned for specific situations are as follows:

Situation	*Error Class*	*Error Code*
During the processing of the request, dynamically allocated memory was not available at an intermediate step so the request cannot be completed.	*RESOURCES*	*OUT_OF_MEMORY*
An unexpected internal error occurred that cannot be recovered from.	*DEVICE*	*INTERNAL_ERROR*
The device is not completely configured and therefore can't fulfill the request.	*DEVICE*	*NOT_CONFIGURED*
Some misconfiguration is preventing the request from being fulfilled.	*DEVICE*	*INCONSISTENT_CONFIGURATION*

[Add to the **BACnetAbortReason** production in Clause **21**, p. 443]

```
BACnetAbortReason ::=       ENUMERATED {
    other                                   (0),
    buffer-overflow                         (1),
    invalid-apdu-in-this-state              (2),
    preempted-by-higher-priority-task       (3),
    segmentation-not-supported              (4),
    …
    window-size-out-of-range                (7),
    application-exceeded-reply-time         (8),
    out-of-resources                        (9),
    tsm-timeout                             (10),
    apdu-too-long                           (11),
    ...
    }
```

[Add to the **Error** production in Clause **21**, p. 445]

```
Error ::= SEQUENCE {

    …
    error-code          ENUMERATED {
                        other                                   (0),
                        abort-apdu-too-long                     (123),
                        abort-application-exceeded-reply-time   (124),
                        abort-buffer-overflow                   (51),
                        abort-invalid-apdu-in-this-state        (52),
                        abort-out-of-resources                  (125),
                        abort-preempted-by-higher-priority-task (53),
                        abort-segmentation-not-supported        (54),
                        abort-proprietary                       (55),
                        abort-other                             (56),
                        abort-tsm-timeout                       (126),
                        abort-window-size-out-of-range          (127),
                        authentication-failed                   (1),
                        ...
                        file-access-denied                      (5),
                        file-full                               (128)
```

incompatible-security-levels (6),
inconsistent-configuration (129),
inconsistent-object-type (130),
inconsistent-parameters (7),
inconsistent-selection-criterion (8),
internal-error (131),
invalid-array-index (42),
…
no-objects-of-specified-type (17),
no-property-specified (77),
no-space-for-object (18),
no-space-to-add-list-element (19),
no-space-to-write-property (20),
not-configured (132),
not-configured-for-triggered-logging (78),
object-deletion-not-permitted (23),
object-identifier-already-exists (24),
operational-problem (25),
optional-functionality-not-supported (45),
out-of-memory (133),
password-failure (26),
…
--see abort-apdu-too-long (123),
--see abort-application-exceeded-reply-time (124),
--see abort-out-of-resources (125),
--see abort-tsm-timeout (126),
--see abort-window-size-out-of-range (127),
--see file-full (128)
--see inconsistent-configuration (129)
--see inconsistent-object-type (130)
--see internal-error (131)
--see not-configured (132)
--see out-of-memory (133)

135-2008*u*-4. Specify proper errors when attempting access to the Log_Buffer property.

Rationale
The standard states that the Log_Buffer may only be read with the ReadRange service. However, in some other services, proper error reporting is not specified.

Addendum 135-2008*u*-4

[Change Clause **15.6.1.1.2**, ReadPropertyConditional service, p. 345]

15.6.1.1.2 List of Property References

This optional parameter, if present, shall be a list of one or more BACnetPropertyReferences, each of which corresponds directly to a specific property of any object selected on the basis of the 'Object Selection Criteria'. Specifying the property ALL indicates that all properties of any selected object shall be returned, including any proprietary properties. Specifying the property REQUIRED means that ~~only~~ *all* those properties having a conformance code of "R" or "W" shall be returned. Specifying the property identifier OPTIONAL means that only those properties that have a conformance code of "O" shall be returned. 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 an Error for that property shall be returned in the List of Read Access Results as specified by Clause 15.6.1.2.1.* If this parameter is absent, no properties shall be returned in the 'List of Read Access Results' portion of the 'Result(+)' primitive; however, the object identifier shall be returned in the 'Object Identifier' parameter.

[Change Clause **15.7.3.2.1**, ReadPropertyMulitple service, p. 351]

15.7.3.1.2 List of Property References

This parameter shall be a list of one or more BACnetPropertyReferences, each of which corresponds directly to a specific property of the object identified above. The property identifier ALL means that all defined properties of the object are to be accessed, including any proprietary properties. The property identifier REQUIRED means that only those properties having a conformance code of "R" or "W" shall be returned. The property identifier OPTIONAL means that only those properties that have a conformance code "O" shall be returned. 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.*

[Add a new entry to **History of Revisions**, p. 688]

(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

Protocol		***Summary of Changes to the Standard***
Version	***Revision***	
…	…	**…**
1	10	**Addendum *u* to ANSI/ASHRAE 135-2008** Approved by the ASHRAE Standards Committee January 23, 2010; by the ASHRAE Board of Directors January 27, 2010; and by the American National Standards Institute January 28, 2010. 1. Clarify the use of RejectPDUs. 2. Add error code UNSUPPORTED_OBJECT_TYPE for CreateObject service. 3. Add new Abort and Error codes. 4. Specify proper Errors when attempting access to the Log_Buffer property.

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.