

BSR/ASHRAE Addendum b
to ANSI/ASHRAE Standard 135-2004

Public Review Draft

ASHRAE® Standard

Proposed Addendum b to Standard 135-2004, *BACnet*®—A *Data Communication Protocol for Building Automation and Control Networks*

**Submitted for Second Public Review
(May 2005)
(Draft Shows Proposed Changes to
Previous Public Review)**

~~This draft has been recommended for public review by the responsible project committee. To submit a comment on this proposed addendum, use the comment form and instructions provided with this draft. The draft is subject to modification until it is approved for publication by the responsible project committee, the ASHRAE Standards Committee, and the Board of Directors. Then it will be submitted to the American National Standards Institute Board of Standards Review (BSR) for approval. Until this time, the current edition of the standard (as modified by any published addenda on the ASHRAE web site) remains in effect. The current edition of any standard may be purchased from the ASHRAE Bookstore @ <http://www.ashrae.org> or by calling 404-636-8400 or 1-800-727-4723 (for orders in the U.S. or Canada).~~

This standard is under continuous maintenance. To propose a change to the current standard, use the change submittal form available on the ASHRAE web site @ <http://www.ashrae.org>.

The appearance of any technical data or editorial material in this public review document does not constitute endorsement, warranty, or guaranty by ASHRAE of any product, service, process, procedure, or design, and ASHRAE expressly disclaims such.

© March 29, 2005. This draft is covered under ASHRAE copyright. Permission to reproduce or redistribute all or any part of this document must be obtained from the ASHRAE Manager of Standards, 1791 Tullie Circle, NE, Atlanta, GA 30329-2305. Phone: 404-636-8400, Ext1125. Fax: 404-321-5478. E-mail: standards.section@ashrae.org.

AMERICAN SOCIETY OF HEATING,
REFRIGERATING AND AIR-CONDITIONING
ENGINEERS, INC.
1791 Tullie Circle, NE · Atlanta GA 30329-2305



[This foreword and the “rationale” on the following page 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 second public review draft of this addendum is to present proposed independent substantive changes 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.

- 135-2004b-1. Add a new Event Log object type, p. 1.
- 135-2004b-2. Add a new Global Group object type, p. 3.
- 135-2004b-3. Add a new Trend Log Multiple object type, p. 5.
- 135-2004b-4. Harmonize the Trend Log object with the new Event Log and Trend Log Multiple objects, p.14.
- 135-2004b-5. Define a means for a device to provide a notification that it has restarted, p. 20.
- 135-2004b-6. Define a means to configure a device to periodically send time synchronization messages, p. 21.
- 135-2004b-7. Extend the number of character sets supported, p. 22.
- 135-2004b-8. Enable devices other than alarm recipients to acknowledge alarms, p. 23.
- 135-2004b-9. Allow MS/TP BACnet Data Expecting Reply frames to be broadcast, p. 24.
- 135-2004b-10. Revise the Clause 5 state machines to handle slow servers, p. 25.
- 135-2004b-11. Add new Error Codes and specify usage, p.26.

Note to Reviewers: In this addendum, changes to the previous public review draft are indicated in the text by *italics* (for additions) and ~~strikethrough~~ (for deletions) unless the instructions specifically mention some other means of indicating the changes. Only these changes are open for review and comment at this time. Additional material is provided for context only and is not open for comment except as it relates to the proposed substantive changes.

135-2004*b*-1. Add a new Event Log object type.

Rationale

There is need for a standard object type to log events.

Addendum 135-2004*b*-1

[Revise **Clauses 12.13.5** through **12.13.7** as shown below, where *italics* indicate addition and ~~strikethrough~~ indicates deletion. **Clause 12.13** was added in the first public review of this addendum.]

12.13.5 Enable

This property, of type BOOLEAN, indicates and controls whether (TRUE) or not (FALSE) logging of events is enabled. ~~A value of FALSE overrides the time interval defined by Start_Time and Stop_Time.~~ *Logging occurs if and only if Enable is TRUE and Local_Time is on or after Start_Time and Local_Time is on or before Stop_Time. If Start_Time contains any wildcards, then it is considered to be equal to 'the start of time'. If Stop_Time contains any wildcards, then it is considered to be equal to 'the end of time'.* Changes in the ~~log's~~ log status are recorded without regard to the value of the Enable property.

12.13.6 Start_Time

This optional property, of type BACnetDateTime, specifies the date and time at or after which logging shall be enabled by this property. If any of the fields of the BACnetDateTime contain "wildcard" values, then the ~~specified time shall be considered invalid and logging shall not occur~~ conditions for logging to be enabled by Start_Time shall be ignored. If Start_Time specifies a date and time after Stop_Time, then logging shall be disabled. ~~If Stop_Time is present, then Start_Time shall also be present.~~ This property shall be writable if present.

12.13.7 Stop_Time

This optional property, of type BACnetDateTime, specifies the date and time at or after which logging shall be disabled by this property. If any of the fields of the BACnetDateTime contain "wildcard" values, then the ~~specified time shall be considered to be invalid and logging shall not occur~~ conditions for logging to be disabled by Stop_Time shall be ignored. If Stop_Time specifies a date and time earlier than Start_Time, then logging shall be disabled. ~~If Stop_Time is present, then Start_Time shall also be present.~~ This property shall be writable if present.

[Renumber **Clauses 12.13.12** through **12.13.26** as **Clauses 12.13.8** through **12.13.22**. These clauses were given incorrect numbering in the first public review draft of this addendum.]

[Revise **Clause 12.13.15**, which was renumbered in the previous instruction, as shown below, where *italics* indicate addition. In the first public review draft, this clause was mistakenly numbered as Clause 12.13.19.]

12.13.15 Last_Notify_Record

This property, of type Unsigned32, represents the SequenceNumber associated with the most recently collected record whose collection *caused the value of the Records_Since_Notification property to become equal to or greater than the value of the Notification_Threshold property* which triggered a notification. If no notification has occurred since logging began, the value of this property shall be zero. This property is required if intrinsic reporting is supported by this object.

[Delete the first-public-review changes shown below in **Table 13.2**, p. 256, and replace them with the revised version of these changes, which is shown immediately below the changes to be deleted.]

[Public review 1 version]

Object Type	Criteria	Event Type
...
Trend Log <i>Trend Log,</i> <i>Event Log</i>	If Event_State is NORMAL and Records_Since_Notification is equal to Notification_Threshold	BUFFER_READY
...

[Public review 2 version]

Object Type	Criteria	Event Type
...
Trend Log <i>Event Log,</i> <i>Trend Log,</i> <i>Trend Log Multiple</i>	If Event_State is NORMAL and Records_Since_Notification is equal to Notification_Threshold	BUFFER_READY
...

[Delete the first-public-review changes shown below in **Table 13.3**, p. 257, and replace them with the revised version of these changes, which is shown immediately below the changes to be deleted.]

[Public review 1 version]

Object	Event Type	Notification Parameters	Referenced Object's Properties
...
Trend Log <i>Trend Log,</i> <i>Event Log</i>	BUFFER_READY	Buffer_Property Previous_Notification Current_Notification	BACnetDeviceObjectPropertyReference ¹ Last_Notify_Record Total_Record_Count
...

¹This parameter conveys a reference to the Log_Buffer property of the ~~Trend Log~~ object.

[Public review 2 version]

Object	Event Type	Notification Parameters	Referenced Object's Properties
...
Trend Log <i>Event Log,</i> <i>Trend Log,</i> <i>Trend Log Multiple</i>	BUFFER_READY	Buffer_Property Previous_Notification Current_Notification	BACnetDeviceObjectPropertyReference ¹ Last_Notify_Record Total_Record_Count
...

¹This parameter conveys a reference to the Log_Buffer property of the ~~Trend Log~~ object.

135-2004b-2. Add a new Global Group object type.

Rationale
 There is need for a standard object type similar to the Group object type except that it can provide a collection of information from objects in a number of BACnet devices and can also deliver that information in an intrinsic event notification when any of the group member objects enters a non-NORMAL state.

Addendum 135-2004b-2

[Revise the following entries in **Table 12-17** as shown below, where *italics* indicate addition and ~~strikethrough~~ indicates deletion. **Table 12-17** was added in the first public review of this addendum.]

Table 12-17. Properties of the Global Group Object Type

Property Identifier	Property Datatype	Conformance Code
...
Present_Value	BACnetARRAY[N] of BACnetPropertyAccessResult	R
Member_Status_Flags	BACnetStatusFlags	R
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
<i>Member_Status_Flags</i>	<i>BACnetStatusFlags</i>	<i>R</i>
...

[Delete the first-public-review version of the new **BACnetPropertyAccessResult** production that was added to **Clause 21**, p.423, in the first public review and replace it the revised version of this addition, which is shown immediately below the version to be deleted.]

[Public review 1 version]

```

BACnetPropertyAccessResult ::= SEQUENCE {
    deviceIdentifier      [0] BACnetObjectIdentifier OPTIONAL
    objectIdentifier     [1] BACnetObjectIdentifier,
    propertyIdentifier    [2] BACnetPropertyIdentifier,
    propertyArrayIndex  [3] Unsigned OPTIONAL,    -- used only with array datatype
                                                    -- if omitted with an array then
                                                    -- the entire array is referenced

    accessResult        CHOICE {
        propertyValue      [4] ABSTRACT-SYNTAX.&Type,
        propertyAccessError [5] Error
    }
}
    
```

[Public review 2 version]

```

BACnetPropertyAccessResult ::= SEQUENCE {
    objectIdentifier     [0] BACnetObjectIdentifier,
    propertyIdentifier    [1] BACnetPropertyIdentifier,
    propertyArrayIndex  [2] Unsigned OPTIONAL,    -- used only with array datatype
                                                    -- if omitted with an array then
                                                    -- the entire array is referenced

    deviceIdentifier     [3] BACnetObjectIdentifier OPTIONAL

    accessResult        CHOICE {
        propertyValue      [4] ABSTRACT-SYNTAX.&Type,
        propertyAccessError [5] Error
    }
}
    
```

[Revise the first-public-review version of **BACnetPropertyIdentifier** production in **Clause 21**, p. 423-428, as shown below, where *italics* indicate addition and ~~strike through~~ indicates deletion. **BACnetPropertyIdentifier** production was revised in the first public review and an additional entry is being revised in the changes shown below.]

```
BACnetPropertyIdentifier ::= ENUMERATED {  
  ...  
  log device object property list (197),  
  logging-type (197),  
  ...  
  see log device object property list (197),  
  -- see logging-type (197),  
  -- see member-status-flags (198),  
  ...  
}
```

- The special property identifiers all, optional, and required are reserved for use in the ReadPropertyConditional and ReadPropertyMultiple services or services not defined in this standard.
-
- Enumerated values 0-511 are reserved for definition by ASHRAE. Enumerated values 512-4194303 may be used by others subject to the procedures and constraints described in Clause 23. ~~The highest enumeration used in this version is 206.~~

135-2004*b*-3. Add a new Trend Log Multiple object type.

Rationale

There is need for a standard object similar to the Trend Log object type but which can record multiple data items in a single record, align its recording intervals to the clock, and to be able to collect the data items upon command (i.e., when a certain property is written).

Addendum 135-2004*b*-3

[Revise **Clause 12.26** as shown below, where *italics* indicate addition and ~~strikethrough~~ indicates deletion. **Clause 12.26** was added in the first public review of this addendum.]

12.26 Trend Log Multiple Object Type

A Trend Log Multiple object monitors one or more properties of one or more referenced objects, either in the same device as the Trend Log Multiple object or in an external device. When predefined conditions are met, the object saves ("logs") the value of the properties and a timestamp ~~in an~~ *into* internal ~~buffer~~ *buffers* for subsequent retrieval. The data may be logged periodically or when a record is "triggered" by a write to the Trigger property. Errors that prevent the acquisition of the data, as well as changes in the status or operation of the logging process itself, are also recorded. Each timestamped buffer entry is called a ~~Trend Log Multiple~~ "record."

Under normal operating conditions, a record shall be added for each monitored property and the timestamp in each of these records shall be the same. If an entry in the Log_DeviceObjectProperty contains a wildcard object or device instance, then no record need be added in the corresponding list. Under certain circumstances, the addition of a record outside of a normal logging operation, such as when the Log_DeviceObjectProperty property is changed, records may be added, or removed from, a subset of the Log_Buffer lists.

Each Trend Log Multiple object maintains an internal, optionally fixed-size, buffer. This buffer fills or grows as log records are added. If the buffer becomes full, the least recent record is overwritten when a new record is added, or collection may be set to stop. Trend Log Multiple records are transferred as ~~BACnetLogMultipleRecords~~ *a list of BACnetLogRecords* using the ReadRange service. The buffer may be cleared by writing a zero to the Record_Count property. Each record in the buffer has an implied SequenceNumber that is equal to the value of the Total_Record_Count property immediately after the record is added.

...

[Revise the following entries in **Table 12-30** as shown below, where *italics* indicate addition and ~~strikethrough~~ indicates deletion. **Table 12-30** was added in the first public review of this addendum.]

Table 12-30. Properties of the Trend Log Multiple Object Type

Property Identifier	Property Datatype	Conformance Code
...
Log_DeviceObjectPropertyList	BACnetDeviceObjectPropertyList	R
<i>Log_DeviceObjectProperty</i>	<i>BACnetARRAY[N] of BACnetDeviceObjectPropertyReference</i>	
<i>Logging_Type</i>	<i>BACnetLoggingType</i>	R
Log_Interval	Unsigned	W
...
Interval_Offset	Unsigned	O ²
Trigger	BOOLEAN	Θ ¹ O
Stop_When_Full	BOOLEAN	R
Buffer_Size	Unsigned32	R
Log_Buffer	<i>BACnetARRAY[N] of List of BACnetLogRecord</i>	R
...

[Delete the first-public-review versions of Clause 12.26.5 through 12.26.8 and replace them with the revised versions of these clauses. For each clause, the versions to be deleted are shown first, followed by the revised version.]

[Public review 1 version]

12.26.5 Enable

This property, of type BOOLEAN, indicates and controls whether (TRUE) or not (FALSE) logging of data is enabled. A value of FALSE overrides the time interval defined by Start_Time and Stop_Time. Changes in the log status are recorded without regard to the value of the Enable property.

[Public review 2 version]

12.26.5 Enable

This property, of type BOOLEAN, indicates and controls whether (TRUE) or not (FALSE) logging of events and collected data is enabled. Logging occurs if and only if Enable is TRUE and Local_Time is on or after Start_Time and Local_Time is on or before Stop_Time. If Start_Time contains any wildcards then it is considered to be equal to 'the start of time'. If Stop_Time contains any wildcards then it is considered to be equal to 'the end of time'. Changes in the log's status are recorded without regard to the value of the Enable property.

[Public review 1 version]

12.26.6 Start_Time

This optional property, of type BACnetDateTime, specifies the date and time at or after which logging shall be enabled by this property. If any of the fields of the BACnetDateTime contain "wildcard" values, the specified time shall be considered to be invalid and logging shall not occur. If Start_Time specifies a date and time after Stop_Time then logging shall be disabled. If Start_Time is present, then Stop_Time shall also be present. This property must be writable if present.

[Public review 2 version]

12.26.6 Start_Time

This optional property, of type BACnetDateTime, specifies the date and time at or after which logging shall be enabled by this property. If any of the fields of the BACnetDateTime contain "wildcard" values, then the conditions for logging to be enabled by Start_Time shall be ignored. If Start_Time specifies a date and time after Stop_Time, then logging shall be disabled. This property shall be writable if present.

When Start_Time is reached, the value of the Enable property is not changed.

[Public review 1 version]

12.26.7 Stop_Time

This optional property, of type BACnetDateTime, specifies the date and time at or after which logging shall be disabled by this property. If any of the fields of the BACnetDateTime contain "wildcard" values, then the specified time shall be considered to be invalid and logging shall not occur. If Stop_Time specifies a date and time earlier than Start_Time then logging shall be disabled. If Stop_Time is present, then Start_Time shall also be present. This property must be writable if present.

[Public review 2 version]

12.26.7 Stop_Time

This optional property, of type BACnetDateTime, specifies the date and time at or after which logging shall be disabled by this property. If any of the fields of the BACnetDateTime contain "wildcard" values, then the conditions for logging to be disabled by Stop_Time shall be ignored. If Stop_Time specifies a date and time earlier than Start_Time, then logging shall be disabled. This property shall be writable if present.

When Stop_Time is reached, the value of the Enable property is not changed.

[Public review 1 version]

12.26.8 Log_DeviceObjectPropertyList

This property, of type BACnetDeviceObjectPropertyList, specifies the Device Identifier and a list of Object Identifiers and Property Identifiers of the properties to be logged.

If this property is writable, it may be restricted to reference only objects inside the device containing the Trend Log Multiple object. If the property is restricted to referencing objects within the containing device, an attempt to write a reference to an object outside the containing device into this property shall cause an Error-PDU to be issued conveying 'error class' = PROPERTY and 'error code' = OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED.

If this property is changed and the Log_Buffer contains any records of type 'log-data', the Log_Buffer shall be purged and a log-multiple-status record specifying 'buffer-purged' shall be stored in the Log_Buffer.

[Public review 2 version]

12.26.8 Log_DeviceObjectProperty

This property, of type BACnetARRAY of BACnetDeviceObjectPropertyReference, specifies the properties to be logged.

If this property is writable, it may be restricted to reference only objects inside the device containing the Trend Log Multiple object. If the property is restricted to referencing objects within the containing device, an attempt to write a reference to an object outside the containing device into this property shall cause an Error-PDU to be issued conveying 'error class' = PROPERTY and 'error code' = OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED.

If this property is changed and the corresponding entry in the Log_Buffer array contains any records of type 'log-data', the Log_Buffer shall be purged and a log-status record specifying 'buffer-purged' shall be stored in the corresponding Log_Buffer entry. It is a local matter whether or not the other Log_Buffer lists are purged as well.

The number of entries in the Log_DeviceObjectProperty property dictates the number of lists in the Log_Buffer array. The i^{th} entry of the Log_Buffer array contains the list of data items for the i^{th} entry in the Log_DeviceObjectProperty array.

[Insert new **Clause 12.26.9** in **Clause 12.26**, which was added in the first public review of this addendum. This addition will result in the renumbering of first-public-review **Clauses 12.26.9** through **12.26.27** as **Clauses 12.26.10** through **12.26.28**. These numbering changes are described in later instructions.]

12.26.9 Logging_Type

This property, of type BACnetLoggingType, specifies whether the Trend Log Multiple collects records using polling or triggered acquisition.

COV Logging is not allowed for a Trend Log Multiple object. If this property is writable, an attempt to write the value 'cov' into this property shall cause an Error-PDU to be issued conveying 'error class' = PROPERTY and 'error code' = VALUE_OUT_OF_RANGE.

[Revise and renumber first-public-review **Clauses 12.26.9** and **12.12.10** as shown below, where *italics* indicate addition and ~~strikethrough~~ indicates deletion.]

12.26.910 Log_Interval

This property, of type Unsigned, specifies the periodic interval in hundredths of seconds for which the referenced properties are to be logged. ~~If this property has the value zero, then periodic logging is disabled and the Trend Log Multiple object shall only acquire data when the value of the Trigger property is changed from FALSE to TRUE.~~

12.26.1011 Align_Intervals

This optional property, of type BOOLEAN, specifies whether (TRUE) or not (FALSE) clock-aligned periodic logging is enabled. If periodic logging is enabled and ~~the logging interval is such that periodic logging~~ *value of Log_Interval* is a factor of (*that is, it divides without remainder*) a second, minute, hour or day, then the beginning of the period specified for logging shall be aligned to the second, minute, hour or day, respectively.

This property has no effect on the behavior of the Trend Log Multiple object if the Logging_Type property has a value other than 'polled'.

[Renumber first-public-review **Clause 12.26.11** as **Clause 12.26.12**. This is the only change to this clause.]

[Revise and renumber first-public-review **Clause 12.26.12** as shown below, where *italics* indicate addition and ~~strikethrough~~ indicates deletion.]

12.26.1213 Trigger

This optional property, of type BOOLEAN, shall cause the Trend Log Multiple object to acquire a log record whenever the value of this property is changed from FALSE to TRUE. It shall remain TRUE while the Trend Log Multiple object is acquiring the data items for a record. When all data items have been collected or it has been determined that all outstanding data requests will not be fulfilled, the Trend Log Multiple object shall reset the value to FALSE.

Writing to the Trigger property is not restricted to network visible write operations; internal processes may control the acquisition of samples by writing to this property.

[Renumber first-public-review **Clauses 12.26.13** and **12.26.14** as **Clauses 12.26.14** and **12.26.15**. This is the only change to these two clauses.]

[Renumber first-public-review **Clauses 12.26.16** through **12.26.19** as **Clauses 12.26.17** through **12.26.20**. This is the only change to these clauses.]

[Revise and renumber first-public-review **Clause 12.26.20** as shown below, where *italics* indicate addition and ~~strikethrough~~ indicates deletion.]

12.26.20/ Last_Notify_Record

This optional property, of type Unsigned32, represents the SequenceNumber associated with the most recently collected record whose collection *caused the value of the Records_Since_Notification property to become equal to or greater than the value of the Notification_Threshold property* which triggered a notification. If no notification has occurred since logging occurred, the value of this property shall be zero. This property is required if intrinsic reporting is supported by this object.

[Renumber first-public-review **Clauses 12.26.21** through **12.26.27** as **Clauses 12.26.22** through **12.26.28**. This is the only change to these clauses.]

[Delete the first-public-review **changes** to **Tables 13-2** and **13-3** and replace them with the following notes in this second public review, as shown below.]

[Public review 1 version]

[Change **Table 13-2**, p.256]

Table 13-2. Standard Objects that May Support Intrinsic Reporting

Object Type	Criteria	Event Type
...		
Trend Log Log, <i>Trend Log Multiple</i>	If Event_State is NORMAL and Records_Since_Notification is equal to Notification_Threshold	BUFFER_READY
...		

[Change **Table 13-3**, p.257]

Table 13-3. Standard Object Property Values Returned in Notifications

Object	Event Type	Notification Parameters	Referenced Object's Properties
...			
Trend Log Log, <i>Trend Log Multiple</i>	BUFFER_READY	Buffer_Property Previous_Notification Current_Notification	BACnetDeviceObjectPropertyReference Previous_Notify_Record Last_Notify_Record
...			

[Public review 2 version]

[Note: Change to **Table 13-2**, p. 256, appears in Addendum 135-2004b-1.]

[Note: Change to **Table 13-3**, p. 257, appears in Addendum 135-2004b-1.]

[Add the following new **BACnetLoggingType** production to **Clause 21**, p.419, for this second public review.]

```

BACnetLoggingType ::= ENUMERATED {
    polled      (0),
    cov         (1),
    triggered   (2),
    ...
}
-- Enumerated values 0-127 are reserved for definition by ASHRAE. Enumerated values
-- 128-255 may be used by others subject to the procedures and constraints described
-- in Clause 23.

```

[Delete **BACnetDeviceObjectPropertyList** production added to **Clause 21**, p.410, in the first public review as shown below, where ~~strikethrough~~ indicates deletion.]

```


BACnetDeviceObjectPropertyList ::= SEQUENCE {
    deviceIdentifier [0] BACnetObjectIdentifier OPTIONAL,
    objectpropertylist [1] SEQUENCE OF SEQUENCE {
        objectIdentifier [2] BACnetObjectIdentifier,
        propertyIdentifier [3] BACnetPropertyIdentifier,
        propertyArrayIndex [4] Unsigned OPTIONAL, used only with array datatype
        if omitted with an array then
        the entire array is referenced
    } OPTIONAL
}


```

[Delete **BACnetLogMultipleRecord** production added to **Clause 21**, p.419, in first public review as shown below, where ~~strikethrough~~ indicates deletion.]

```


BACnetLogMultipleRecord ::= SEQUENCE {
    timestamp [0] BACnetDateTime,
    logMultipleRecord [1] CHOICE {
        logStatus [0] BACnetLogStatus,
        failure [1] Error,
        timeChange [2] REAL,
        logData [3] SEQUENCE OF ReadAccessResult
    }
}


```

[Delete the first-public-review changes shown below to **Clause 22.2.1.4**, p. 464, and replace them with the revised version of these changes, which is shown immediately below the changes to be deleted.]

[Public review 1 version]

22.2.1.4 Trending

"Trending" is the accumulation of (time, value) *or (time, list of value)* pairs at specified rates for a specified duration. The values are those of a specific ~~property~~ *properties* of a specific ~~object~~ *objects*. "Trending" is distinguished from the real-time plotting of data in that the data are usually destined for long-term storage and the sampling intervals are usually longer. Interoperability in this area permits the establishment of ~~trending~~ *logging* parameters and the subsequent retrieval and storage of ~~trend~~ *logged* data.

[Public review 2 version]

22.2.1.4 Trending

"Trending" is the accumulation of (time, value) pairs at specified rates for a specified duration. The values are those of a specific ~~property~~ *properties* of a specific ~~object~~ *objects*. "Trending" is distinguished from the real-time plotting of data in that the data are usually destined for long-term storage and the sampling intervals are usually longer. Interoperability in this area permits the establishment of ~~trending~~ *logging* parameters and the subsequent retrieval and storage of ~~trend~~ *logged* data.

[Add the following new entry to **Table 23-1**, p. 437, for this second public review, where *italics* indicate addition. This table was not changed in the first public review.]

Table 23-1. Extensible Enumerations

Enumeration Name	Reserved Range	Maximum Value
...
BACnetLifeSafetyOperation	0-63	65535
<i>BACnetLoggingType</i>	<i>0-63</i>	<i>255</i>
BACnetMaintenance	0-255	65535
...

[Revise the following first-public-review addition to **Annex C**, p. 464, as shown below, where *italics* indicate addition and ~~strikethrough~~ indicates deletion. Only the revised parts of the addition are shown here.]

TREND-LOG-MULTIPLE ::= SEQUENCE {

- ...
- stop-time [143] BACnetDateTime OPTIONAL,
- ~~log-device-object-property-list~~ [132] ~~BACnetDeviceObjectPropertyList,~~
- log-device-object-property* [132] *SEQUENCE OF BACnetDeviceObjectProperty,*
--accessed as a BACnetARRAY
- logging-type* [197] *BACnetLoggingType,*
- log-interval [134] Unsigned,
- align-intervals [193] BOOLEAN OPTIONAL,
- interval-offset [195] Unsigned OPTIONAL,
- trigger [205] BOOLEAN OPTIONAL,
- stop-when-full [144] BOOLEAN,
- buffer-size [126] Unsigned32,
- ~~log-buffer~~ [131] ~~SEQUENCE OF BACnetLogRecord,~~
- log-buffer* [131] *SEQUENCE OF SEQUENCE OF BACnetLogRecord,*
--accessed as a BACnetARRAY
- record-count [141] Unsigned32,
- ...

}

[Revise the following first-public-review addition to **Annex D**, p. 484, as shown below, where *italics* indicate addition and ~~strikethrough~~ indicates deletion. Only the revised parts of **Annex D.26**, which was added in the first public review, are shown here.]

D.26 Example of a Trend Log Multiple Object

The following is an example of a Trend Log Multiple object that logs data every 5 minutes from objects in remote device 100 and which performs buffer-ready notification via intrinsic reporting.

```
Property: Object_Identifier = (Trend Log Multiple, Instance 1)
...
Property: Enable = TRUE
Property: Log_DeviceObjectPropertyList = ((Device, Instance 100), ((Analog Input, Instance 3, Present_Value),
(Analog Input, Instance 3, Status_Flags), (Binary Output, Instance 5,
Present_Value))
Property: Log_DeviceObjectProperty = ((Analog Input, Instance 3), Present_Value, (Device, Instance 100)),
((Analog Input, Instance 3), Status_Flags, (Device, Instance 100)),
((Binary Output, Instance 5), Present_Value, (Device, Instance 100))
Property: Logging_Type = POLLED
Property: Log_Interval = 30000
...
Property: Records_Since_Notification = 30
Property: Previous_Notify_Record = 130927
Property: Last_Notify_Record = 131010
...
```

[Revise the following first-public-review addition to **Annex K.4.8**, p. 583, as shown below, where *italics* indicate addition and ~~strikethrough~~ indicates deletion. **Annex K.4.8** through **K.4.10** was added in the first public review.]

K.4.8 BIBB - Trending-Viewing and Modifying Multiple Values External-B (T-VMMV-E-B)

The B device is capable of logging multiple properties of multiple objects contained in other devices. The B device shall support T-VMMV-I-B and DS-RPM-A. The Log_Interval and ~~Log_DeviceObjectPropertyList~~ *Log_DeviceObjectProperty* properties must be writable.

135-2004*b*-4. Harmonize the Trend Log object with the new Event Log and Trend Log Multiple objects.

Rationale

Several features were added in the Event Log and Trend Log Multiple object types, along with some changes in the language. These features and language are added to the Trend Log object to make it consistent with the other object types.

Addendum 135-2004*b*-4

[Revise **Clause 12.25** as shown below, where *italics* indicate addition and ~~strikethrough~~ indicates deletion. Other parts of **Clause 12.25** were revised in the first public review but not the part shown below.]

A Trend Log object monitors a property of a referenced object and, when predefined conditions are met, saves ("logs") the value of the property and a timestamp in an internal buffer for subsequent retrieval. The data may be logged periodically ~~or~~, upon a change of value *or when "triggered" by a write to the Trigger property*. Errors that prevent the acquisition of the data, as well as changes in the status or operation of the logging process itself, are also recorded. Each timestamped buffer entry is called a trend log "record."

[Delete the first-public-review changes shown below to **Table 12-29**, p. 247, and replace them with the revised version of these changes, which is shown immediately below the changes to be deleted.]

[Public review 1 version]

Table 12-29. Properties of the Trend Log Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Log_Enable <i>Enable</i>	BOOLEAN	W
...

[Public review 2 version]

Table 12-29. Properties of the Trend Log Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Log_Enable <i>Enable</i>	BOOLEAN	W
...
<i>Logging_Type</i>	<i>BACnetLoggingType</i>	<i>R</i>
Log_Interval	Unsigned	O ^{1,2}
<i>Align_Intervals</i>	<i>BOOLEAN</i>	<i>O</i> ³
<i>Interval_Offset</i>	<i>Unsigned</i>	<i>O</i> ³
...
Client_COV_Increment	BACnetClientCOV	O
<i>Trigger</i>	<i>BOOLEAN</i>	<i>O</i>
Stop_When_Full	BOOLEAN	R
...
Notification_Threshold	Unsigned32	$\Theta^3 O^4$
Records_Since_Notification	Unsigned32	$\Theta^3 O^4$
Last_Notify_Record	Unsigned32	$\Theta^3 O^4$
Event_State	BACnetEventState	R
Notification_Class	Unsigned	$\Theta^3 O^4$
Event_Enable	BACnetEventTransitionBits	$\Theta^3 O^4$
Acked_Transitions	BACnetEventTransitionBits	$\Theta^3 O^4$
Notify_Type	BACnetNotifyType	$\Theta^3 O^4$
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	$\Theta^3 O^4$
...

¹ These properties are required to be present if the monitored property is a BACnet property.

² If present, these properties are required to be writable.

³ These properties are required to be present if the object supports clock-aligned logging.

³⁴ These properties are required to be present if the object supports intrinsic reporting.

[Delete the first-public-review changes shown below to **Clauses 12.25.5** through **12.25.7**, p. 248, and replace them with the revised version of these changes, which is shown immediately below the changes to be deleted.]

[Public review 1 version]

12.25.5 ~~Log_Enable~~ Enable

This property, of type BOOLEAN, indicates and controls whether (TRUE) or not (FALSE) logging *of events and collected data* is enabled. A value of FALSE overrides the time interval defined by Start_Time and Stop_Time. Changes in the log status are recorded without regard to the value of the Enable property.

[Public review 2 version]

12.25.5 ~~Log_Enable~~ Enable

This property, of type BOOLEAN, indicates and controls whether (TRUE) or not (FALSE) logging *of events and collected data* is enabled. ~~A value of FALSE overrides the time interval defined by Start_Time and Stop_Time. If logging is otherwise enabled by the Start_Time and Stop_Time properties, changes to the value of the Log_Enable property shall be recorded in the log. When the device begins operation the value TRUE shall be recorded in the log.~~ *Logging occurs if and only if Enable is TRUE and Local_Time is on or after Start_Time and Local_Time is on or before Stop_Time. If Start_Time contains any wildcards then it considered to be equal to 'the start of time'. If Stop_Time contains any wildcards then it is considered to be equal to 'the end of time'.* Changes in the log's status are recorded without regard to the value of the Enable property.

[Public review 1 version]

12.25.6 Start_Time

This *optional* property, of type BACnetDateTime, specifies the date and time at or after which logging shall be enabled by this property. If any of the fields of the BACnetDateTime contain "wildcard" values, ~~then the conditions for logging to be enabled by Start_Time shall be ignored.~~ *then the specified time shall be considered to be invalid and logging shall not occur.* If Start_Time specifies a date and time after Stop_Time, then logging shall be disabled. This property ~~must~~ *shall* be writable if present.

[Public review 2 version]

12.25.6 Start_Time

This *optional* property, of type BACnetDateTime, specifies the date and time at or after which logging shall be enabled by this property. If any of the fields of the BACnetDateTime contain "wildcard" values, then the conditions for logging to be enabled by Start_Time shall be ignored. If Start_Time specifies a date and time after Stop_Time, then logging shall be disabled. This property ~~must~~ *shall* be writable if present.

When Start_Time is reached, the value of the Enable property is not changed.

[Public review 1 version]

12.25.7 Stop_Time

This *optional* property, of type BACnetDateTime, specifies the date and time at or after which logging shall be disabled by this property. If any of the fields of the BACnetDateTime contain "wildcard" values, ~~then the conditions for logging to be enabled by Stop_Time shall be ignored.~~ *then the specified time shall be considered to be invalid and logging shall not occur.* If Stop_Time specifies a date and time earlier than Start_Time, then logging shall be disabled. This property ~~must~~ *shall* be writable if present.

[Public review 2 version]

12.25.7 Stop_Time

This *optional* property, of type BACnetDateTime, specifies the date and time at or after which logging shall be disabled by this property. If any of the fields of the BACnetDateTime contain "wildcard" values, then the conditions for logging to be ~~enabled~~ *disabled* by Stop_Time shall be ignored. If Stop_Time specifies a date and time earlier than Start_Time, then logging shall be disabled. This property ~~must~~ *shall* be writable if present.

When Stop_Time is reached, the value of the Enable property is not changed.

[There are no changes to **Clause 12.25.8** for this second public review. It was revised in the first public review.]

[Insert new **Clause 12.25.9**, p.248, for this second public review. This addition will result in the renumbering of subsequent clauses, which is described in the instructions that follow.]

12.25.9 Logging_Type

This property, of type BACnetLoggingType, specifies whether the Trend Log collects records using polling, COV or triggered acquisition.

If this property is writable, an attempt to write a value not supported by the object into this property shall cause an Error-PDU to be issued conveying 'error class' = PROPERTY and 'error code' = OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED.

If this property has the value 'cov' then the Trend Log shall issue COV subscriptions for the referenced property, and shall log the COV notifications if they indicate a changed value.

If this property has the value 'polled' then the Trend Log shall periodically poll the monitored property on the interval defined by the Log_Interval, Align_Intervals, and Interval_Offset properties.

[Renumber **Clause 12.25.9** as **Clause 12.25.10** and then delete the first-public-review changes shown below to this clause, p. 248, and replace them with the revised version of these changes, which is shown immediately below the changes to be deleted. Note that this clause was mislabeled as **Log_DeviceObjectProperty** in the first public review.]

[Public review 1 version]

12.25.9 Log_DeviceObjectProperty

This *optional* property, of type Unsigned, specifies the periodic interval in hundredths of seconds for which the referenced property is to be logged. If this property has the value zero then the Trend Log shall issue COV subscriptions for the referenced property. The value of this property ~~must~~ shall be non-zero if COV_Resubscription_Interval is not present. This property ~~must~~ shall be writable if present.

[Public review 2 version]

12.25.910 Log_Interval

This *optional* property, of type Unsigned, specifies the periodic interval in hundredths of seconds for which the referenced property is to be logged. ~~If this property has the value zero then the Trend Log shall issue COV subscriptions for the referenced property.~~ The value of this property ~~must~~ shall be non-zero if COV_Resubscription_Interval is not present. This property ~~must~~ shall be writable if present.

[Insert new **Clauses 12.25.11** and **12.25.12**, p.248, for this second public review. This addition will result in the renumbering of subsequent clauses, which is described in the instructions that follow.]

12.25.11 Align_Intervals

This optional property, of type BOOLEAN, specifies whether (TRUE) or not (FALSE) clock-aligned periodic logging is enabled. If clock-aligned periodic logging is enabled and the value of Log_Interval is a factor of (that is, divides without remainder) a second, minute, hour or day, then the beginning of the period specified for logging shall be aligned to the second, minute, hour or day, respectively.

This property has no effect on the behavior of the Trend Log object if the Logging_Type property has a value other than 'polled'.

12.25.12 Interval_Offset

This optional property, of type Unsigned, specifies the offset in hundredths of seconds from the beginning of the period specified for logging until the actual acquisition of a log record begins. The offset used shall be the value of Interval_Offset modulo the value of Log_Interval; i.e., if Interval_Offset has the value 31 and Log_Interval is 30, the offset used shall be 1. Interval_Offset shall have no effect if Align_Intervals = FALSE.

[Renumber Clauses 12.25.10 and 12.25.11 as Clauses 12.25.13 and 12.25.14. This is the only change to these clauses.]

[Insert new Clause 12.25.15, p.248, for this second public review. This addition will result in the renumbering of subsequent clauses, which is described in the instructions that follow.]

12.25.15 Trigger

This optional property, of type BOOLEAN, shall cause the Trend Log object to acquire a log record whenever the value of this property is changed from FALSE to TRUE. It shall remain TRUE while the Trend Log object is acquiring the data items for a record. When all data items have been collected or it has been determined that all outstanding data requests will not be fulfilled, the Trend Log object shall reset the value to FALSE.

[Renumber Clauses 12.25.12 and 12.25.13 as Clauses 12.25.16 and 12.25.17. This is the only change to these clauses.]

[Renumber Clause 12.25.14 as Clause 12.25.18 and then delete the first-public-review changes shown below to this clause, p. 249, and replace them with the revised version of these changes, which is shown immediately below the changes to be deleted.]

[Public review 1 version]

12.25.14 Log_Buffer

...
The choices available for the LogDatum are listed below:

...
buffer-purged ~~This flag shall be set to TRUE whenever the buffer is deleted by a write of the value zero to the Record_Count property. This flag shall be set to TRUE whenever the buffer is cleared by writing zero to the Record_Count property or by a change to the Log_DeviceObjectPropertyList property. After this value is recorded in the buffer, the subsequent immediate change to FALSE shall not be recorded.~~

[Public review 2 version]

12.25.14/8 Log_Buffer

...
The choices available for the LogDatum are listed below:

...
log-disabled ~~This flag is set whenever the Trend Log object is disabled, such as when Log_Enable is set to FALSE. Whenever the Trend Log object begins operation, this flag shall be presumed to have changed from TRUE to FALSE and a log entry shall be made.~~
This flag is changed whenever collection of records by the Trend Log object is enabled or disabled. It shall be TRUE if Enable is FALSE, or the local time is outside the range defined by Start_Time and Stop_Time, or the addition of this record will cause the buffer to be full and Stop_When_Full is TRUE; otherwise it shall be FALSE.

[Renumber **Clauses 12.25.15** through **12.25.18** as **Clauses 12.25.19** through **12.25.22**. This is the only change to these clauses.]

[Renumber **Clause 12.25.19** as **Clause 12.25.23**, p. 250, and revise it as shown below. This clause was not changed in the first public review.]

12.25.1923 Last_Notify_Record

This property, of type Unsigned32, represents the SequenceNumber associated with the most recently collected record whose collection *caused the value of the Records_Since_Notification property to become equal to or greater than the value of the Notification_Threshold property* which triggered a notification. If no notification has occurred since logging began the value of this property shall be zero. This property is required if intrinsic reporting is supported by this object.

[Renumber **Clauses 12.25.20** through **12.25.26** as **Clauses 12.25.24** through **12.25.30**. This is the only change to these clauses.]

[Add the following notes for this second public review.]

[Note: Addition of BACnetLoggingType production, **Clause 21**, appears in Addendum 135-2004b-3.]

[Note: Addition of BACnetLoggingType to **Table 23-1**, appears in Addendum 135-2004b-3.]

[Revise **Annex C**, p. 464, as shown below, where *italics* indicate addition. This change is in addition to those shown in the first public review draft.]

```
TREND-LOG ::= SEQUENCE {  
...  
logging-type           [197] BACnetLoggingType,  
start-time           [142] BACnetDateTime OPTIONAL,  
...  
}
```

[Revise **Annex D.25**, p. 483, as shown below, where *italics* indicate addition. The changes shown are in addition to those shown in the first public review draft.]

D.25 Example of a Trend Log Object

The following is an example of a Trend Log object that periodically logs data from an object in a remote device and which performs buffer-ready notification via intrinsic reporting.

```
...  
Property: Logging_Type = POLLED  
Property: Log_Interval = 6,000  
Property: Align_Intervals = FALSE  
Property: Interval_Offset = 0  
Property: Trigger = FALSE  
...
```

135-2004b-5. Define a means for a device to provide a notification that it has restarted.

Rationale

When a BACnet device restarts, it could lose some of its configuration and subscriptions. Other devices may depend on this configuration or subscription information for change of value notifications or other purposes. This new restart procedure provides a means to notify peer devices that a restart has occurred, enabling them to take appropriate action.

Addendum 135-2004b-5

[Change **BACnetPropertyStates** production in **Clause 21**, p. 428, for this second public review as shown below, where *italics* indicate addition. This part of the standard was not changed in the first public review.]

BACnetPropertyStates ::= CHOICE {

- This production represents the possible datatypes for properties that
- have discrete or enumerated values. The choice must be consistent with the
- datatype of the property referenced in the Event Enrollment Object.

```
    boolean-value      [0] BOOLEAN,  
    ...  
    life-safety-state  [13] BACnetLifeSafetyState,  
    restart-reason     [14] BACnetRestartReason,  
    ...  
}
```


135-2004b-6. Define a means to configure a device to periodically send time synchronization messages.

Rationale
 There is need for an interoperable means for configuring a device to periodically send TimeSynchronization and UTCTimeSynchronization messages.

Addendum 135-2004b-6

[Delete the first-public-review changes shown below to **Table 12-13**, p. 178, and replace them with the revised version of these changes, which is shown immediately below the changes to be deleted. The additional changes in this second public review only affect footnote numbering.]

[Public review 1 version]

Table 12-13. Properties of the Device Object Type

Property Identifier	Property Datatype	Conformance Code
...
Time_Synchronization_Recipients	List of BACnetRecipient	O ⁵
UTC_Time_Synchronization_Recipients	List of BACnetRecipient	O ⁵
Time_Synchronization_Interval	Unsigned	O ⁶
Align_Intervals	BOOLEAN	O ⁶
Interval_Offset	Unsigned	O ⁶
...

⁵ ~~Required if PICS indicates that this device is a Time Master.~~ If this property is present, then Time_Synchronization_Interval, Align_Intervals and Interval_Offset shall be present. If present, this property shall be writable.

⁶ If either Time_Synchronization_Recipients or UTC_Time_Synchronization_Recipients is present then, this property shall be present and writable.

...

[Public review 2 version]

Table 12-13. Properties of the Device Object Type

Property Identifier	Property Datatype	Conformance Code
...
Time_Synchronization_Recipients	List of BACnetRecipient	O ⁵
UTC_Time_Synchronization_Recipients	List of BACnetRecipient	O ⁵
Time_Synchronization_Interval	Unsigned	O ¹³
Align_Intervals	BOOLEAN	O ¹³
Interval_Offset	Unsigned	O ¹³
...

⁵ ~~Required if PICS indicates that this device is a Time Master.~~ If this property is present, then Time_Synchronization_Interval, Align_Intervals and Interval_Offset shall be present. If present, this property shall be writable.

¹³ If either Time_Synchronization_Recipients or UTC_Time_Synchronization_Recipients is present, then this property shall be present and writable.

...

135-2004b-7. Extend the number of character sets supported.

Rationale

In Europe and Asia there is need for a greater number of character sets than are currently supported by BACnet. The name of the character set JIS C 6226 is also brought current to JIS X 0208.

Addendum 135-2004b-7

[This part of Addendum b, Addendum 135-2004b-7, has not changed since the first public review.]

135-2004b-8. Enable devices other than alarm recipients to acknowledge alarms.

Rationale

Devices that are not in the list of alarm recipients are currently unable to acknowledge alarms. This prevents workstations that have learned about an alarm through other means from acknowledging it.

Addendum 135-2004b-8

[This part of Addendum b, Addendum 135-2004b-8, has not changed since the first public review.]

135-2004b-9. Allow MS/TP BACnet Data Expecting Reply frames to be broadcast.

Rationale

The network layer allows a device to broadcast on its local LAN a message to be routed to a device on some other network (see Clause 6.5.3), but the MS/TP Master Node state machine does not permit an MS/TP router to receive such a message. This addendum changes the state machine so that the MS/TP router will receive and process broadcast BACnet Data Expecting Reply frames.

Addendum 135-2004b-9

[This part of Addendum b, Addendum 135-2004b-9, has not changed since the first public review.]

135-2004b-10. Revise the Clause 5 state machines to handle slow servers.

Rationale

A sequence of events was discovered that would cause a segmented request to fail:

- Client sends a segmented request and receives SegmentAck for each segment sent.
- After sending the final SegmentAck, server processes the request, taking a very long time.
- Client times out and begins re-sending the entire request.
- If server sees a re-sent segment, it takes the AWAIT_RESPONSE:DuplicateSegmentReceived transition and sends SegmentAck with sequence number equal to LastSequenceNumber.
- When client receives the SegmentAck, it takes the Segmented_REQUEST:DuplicateACK_Received transition and ignores the SegmentAck.
- When server finishes processing, it sends a SimpleACK (or other confirmed) response.
- When client receives the response, it takes the UnexpectedPDU_Received transition, sends an Abort and ends the transaction.

The proposed change revises the Clause 5 state machines to handle this situation.

Addendum 135-2004b-10

[This part of Addendum b, Addendum 135-2004b-10, has not changed since the first public review.]

135-2004b-11. Add new Error Codes and specify usage.

Rationale

A comprehensive set of reviews has shown the need for additional error classes and codes to accurately convey the error situation being reported.

Addendum 135-2004b-11

[Delete the first-public-review changes shown below to the **Error** production, pp. 406-207, and replace them with the revised version of these changes, which is shown immediately below the changes to be deleted.]

[Public review 1 version]

Error ::= SEQUENCE {

-- NOTE: The valid combinations of error-class and error-code are defined in Clause 18.

```

error-class  ENUMERATED {
    device      (0),
    object      (1),
    property    (2),
    resources   (3),
    security    (4),
    services    (5),
    vt          (6),
    communication (7),
    ...
},

```

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

error-code ENUMERATED {

```

    other (0),
    abort-buffer-overflow (50),
    abort-invalid-apdu-in-this-state (51),
    abort-preempted-by-higher-priority-task (52),
    abort-segmentation-not-supported (53),
    abort-proprietary (54),
    abort-other (55),
    authentication-failed (1),
    ...
    invalid-parameter-data-type (13),
    invalid-tag, (56)
    invalid-time-stamp (14),
    key-generation-error (15),
    missing-required-parameter (16),
    network-down (57),
    no-objects-of-specified-type (17),
    ...
    read-access-denied (27),
    reject-buffer-overflow (58),
    reject-inconsistent-parameters (59),
    reject-invalid-parameter-data-type (60),
    reject-invalid-tag (61),
    reject-missing-required-parameter (62),
    reject-parameter-out-of-range (63),

```

<i>reject-too-many-arguments</i>	(64),
<i>reject-undefined-enumeration</i>	(65),
<i>reject-unrecognized-service</i>	(66),
<i>reject-proprietary</i>	(67),
<i>reject-other</i>	(68),
security-not-supported	(28),
service-request-denied	(29),
timeout	(30),
<i>unknown-device</i>	(69),
unknown-object	(31),
unknown-property	(32),
<i>unknown-route</i>	(70),
-- this enumeration was removed	(33),
unknown-vt-class	(34),
unknown-vt-session	(35),
unsupported-object-type	(36),
<i>value-not-initialized</i>	(71),
value-out-of-range	(37),
...	
-- see <i>invalid-configuration-data</i>	(46),
-- see <i>datatype-not-supported</i>	(47),
-- see <i>abort-buffer-overflow</i>	(50),
-- see <i>abort-invalid-apdu-in-this-state</i>	(51),
-- see <i>abort-preempted-by-higher-priority-task</i>	(52),
-- see <i>abort-segmentation-not-supported</i>	(53),
-- see <i>abort-proprietary</i>	(54),
-- see <i>abort-other</i>	(55),
-- see <i>invalid-tag</i>	(56)
-- see <i>network-down</i>	(57),
-- see <i>reject-buffer-overflow</i>	(58),
-- see <i>reject-inconsistent-parameters</i>	(59),
-- see <i>reject-invalid-parameter-data-type</i>	(60),
-- see <i>reject-invalid-tag</i>	(61),
-- see <i>reject-missing-required-parameter</i>	(62),
-- see <i>reject-parameter-out-of-range</i>	(63),
-- see <i>reject-too-many-arguments</i>	(64),
-- see <i>reject-undefined-enumeration</i>	(65),
-- see <i>reject-unrecognized-service</i>	(66),
-- see <i>reject-proprietary</i>	(67),
-- see <i>reject-other</i>	(68),
-- see <i>unknown-device</i>	(69),
-- see <i>unknown-router</i>	(70),
-- see <i>value-not-initialized</i>	(71),

...
}
-- Enumerated values 0-255 are reserved for definition by ASHRAE. Enumerated values
-- 256-65535 may be used by others subject to the procedures and constraints described
-- in Clause 23. The last enumeration used in this version is ~~47~~ 71.

}

[Public review 2 version]

Error ::= SEQUENCE {

-- NOTE: The valid combinations of error-class and error-code are defined in Clause 18.

```

error-class  ENUMERATED {
    device      (0),
    object      (1),
    property    (2),
    resources   (3),
    security    (4),
    services    (5),
    vt          (6),
    communication (7),
    ...
},

```

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

```

error-code  ENUMERATED {
    other (0),
    abort-buffer-overflow (51),
    abort-invalid-apdu-in-this-state (52),
    abort-preempted-by-higher-priority-task (53),
    abort-segmentation-not-supported (54),
    abort-proprietary (55),
    abort-other (56),
    authentication-failed (1),
    ...
    invalid-parameter-data-type (13),
    invalid-tag (57),
    invalid-time-stamp (14),
    key-generation-error (15),
    missing-required-parameter (16),
    network-down (58),
    no-objects-of-specified-type (17),
    ...
    read-access-denied (27),
    reject-buffer-overflow (59),
    reject-inconsistent-parameters (60),
    reject-invalid-parameter-data-type (61),
    reject-invalid-tag (62),
    reject-missing-required-parameter (63),
    reject-parameter-out-of-range (64),
    reject-too-many-arguments (65),
    reject-undefined-enumeration (66),
    reject-unrecognized-service (67),
    reject-proprietary (68),
    reject-other (69),
    security-not-supported (28),
    service-request-denied (29),
    timeout (30),
    unknown-device (70),
    unknown-object (31),
    unknown-property (32),

```

```
        unknown-route (71),
        -- this enumeration was removed (33),
        unknown-vt-class (34),
        unknown-vt-session (35),
        unsupported-object-type (36),
        value-not-initialized (72),
        value-out-of-range (37),
    ...
    -- see duplicate-object-id (49),
    -- see property-is-not-an-array (50),
    -- see abort-buffer-overflow (51),
    -- see abort-invalid-apdu-in-this-state (52),
    -- see abort-preempted-by-higher-priority-task (53),
    -- see abort-segmentation-not-supported (54),
    -- see abort-proprietary (55),
    -- see abort-other (56),
    -- see invalid-tag (57),
    -- see network-down (58),
    -- see reject-buffer-overflow (59),
    -- see reject-inconsistent-parameters (60),
    -- see reject-invalid-parameter-data-type (61),
    -- see reject-invalid-tag (62),
    -- see reject-missing-required-parameter (63),
    -- see reject-parameter-out-of-range (64),
    -- see reject-too-many-arguments (65),
    -- see reject-undefined-enumeration (66),
    -- see reject-unrecognized-service (67),
    -- see reject-proprietary (68),
    -- see reject-other (69),
    -- see unknown-device (70),
    -- see unknown-route (71),
    -- see value-not-initialized (72),
    ...
```

```
}
-- Enumerated values 0-255 are reserved for definition by ASHRAE. Enumerated values
-- 256-65535 may be used by others subject to the procedures and constraints described
-- in Clause 23. The last enumeration used in this version is 47.
```

```
}
```