

ANSI/ASHRAE Addendum n
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).

© Copyright 2010 American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.

ISSN 1041-2336



**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, <i>Chair*</i>	Daniel P. Giorgis	John J. Lynch
Carl Neilson, <i>Vice-Chair</i>	David G. Holm berg	Carl J. Ruther
Sharon E. Dinges, <i>Secretary*</i>	Bernhard Isler*	David G. Shike
Donald P. Alexander*	Robert L. Johnson	Ted Sunderland
Barry B. Bridges*	Stephen Karg*	William O. Swan, III
Coleman L. Brumley, Jr.*	Simon Lemaire	David B. Thompson*
Ernest C. Bryant	J. Damian Ljungquist*	Daniel A Traill
James F. Butler	James G. Luth	Stephen J. Treado*
A.J. Capowski		J . Michael Whitcomb*
Craig P. Gemmill		David F. White

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

ASHRAE STANDARDS COMMITTEE 2009–2010

Steven T. Bushby, <i>Chair</i>	Merle F. McBride
H. Michael Newman, <i>Vice-Chair</i>	Frank Myers
Robert G. Baker	Janice C. Peterson
Michael F. Beda	Douglas T. Reindl
Hoy R. Bohanon, Jr.	Lawrence J. Schoen
Kenneth W. Cooper	Boggarm S. Setty
K. William Dean	Bodh R. Subherwal
Martin Dieryckx	James R. Tauby
Allan B. Fraser	James K. Vallort
Katherine G. Hammack	William F. Walter
Nadar R. Jayaraman	Michael W. Woodford
Byron W. Jones	Craig P. Wray
Jay A. Kohler	Wayne R. Reedy, <i>BOD ExO</i>
Carol E. Marriott	Thomas E. Watson, <i>CO</i>

Stephanie Reiniche, *Manager of Standards*

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.

DISCLAIMER

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

ASHRAE INDUSTRIAL ADVERTISING POLICY ON STANDARDS

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

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

[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

Addendum 135*n* 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.

135-2008*n*-1. Add support for long Backup and Restore preparation times, p. 2.

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 ~~strike through~~. Where entirely new subclauses are added, plain type is used throughout.

135-2008n-1. Add support for long Backup and Restore preparation times.

Rationale
 In larger devices, preparations to perform Clause 19.1 Backup and Restore operations can take a considerable amount of time, much greater than (say) typical APDU_Timeout values. A mechanism for supporting such devices is needed.

Addendum 135-2008n-1

[Change **Table 12-13**, p. 180]

Table 12-13. Properties of the Device Object Type

Property Identifier	Property Datatype	Conformance Code
...
Database_Revision	Unsigned	R
Configuration_Files	BACnetARRAY[N] of BACnetObjectIdentifier	O ⁷
Last_Restore_Time	BACnetDateTime	O ⁷
Backup_Failure_Timeout	Unsigned16	O ⁸
Backup_Preparation_Time	Unsigned16	O
Restore_Preparation_Time	Unsigned16	O
Restore_Completion_Time	Unsigned16	O
Backup_And_Restore_State	BACnetBackupState	O
...

⁷ These properties are required if the device supports *execution* of the backup and restore procedures.

⁸ This property ~~must~~ *shall* be present and writable if the device supports the backup and restore procedures.

[Change Clause **12.11.35**, p. 184]

12.11.35 Database_Revision

This property, of type Unsigned, is a logical revision number for the device's database. It is incremented when an object is created, an object is deleted, an object's name is changed, an object's Object_Identifier property is changed, or a restore is performed *with the exception that the creation and deletion of temporary configuration files during a backup or restore procedure shall not affect this property.*

[Add new Clauses **12.11.X1** through **12.11.X4**, p. 187]

12.11.X1 Backup_Preparation_Time

This optional property, of type Unsigned16, indicates the amount of time in seconds that the device might remain unresponsive after the sending of a ReinitializeDevice-ACK at the start of a backup procedure. The device that initiated the backup shall either wait the period of time specified by this property or be prepared to encounter communication timeouts during this period. The use of this value is described in more detail in Clause 19.1.

This property is required if the device supports execution of the backup and restore procedures and cannot complete backup preparation within the minimum value it will accept in its APDU_Timeout property.

12.11.X2 Restore_Preparation_Time

This optional property, of type Unsigned16, indicates the amount of time in seconds that the device is allowed to remain unresponsive after the sending of a ReinitializeDevice-ACK at the start of a restore procedure. The restoring device shall either wait or be prepared to encounter communication timeouts during this period. The use of this value is described in more detail in Clause 19.1.

This property is required if the device supports execution of the backup and restore procedures and cannot complete restore preparation within the minimum value it will accept in its APDU_Timeout property.

12.11.X3 Restore_Completion_Time

This optional property, of type Unsigned16, indicates the amount of time in seconds that the device is allowed to remain unresponsive after the sending of a ReinitializeDevice-ACK at the end of a restore procedure. The restoring device shall either wait or be prepared to encounter communication timeouts during this period. The use of this value is described in more detail in Clause 19.1.

This property is required if the device supports execution of the backup and restore procedures and cannot respond to subsequent communications within the minimum value it will accept in its APDU_Timeout property.

12.11.X4 Backup_And_Restore_State

This optional property, of type BACnetBackupState, indicates a server device's backup and restore state. The use of this value is described in more detail in Clause 19.1.

[Change Clause **12.13.6**, p. 193]

12.13.6 File_Size

This property, of type Unsigned, indicates the size of the file data in octets. If the size of the file can be changed by writing to the file, and File_Access_Method is STREAM_ACCESS, then this property shall be writable.

Writing to the File_Size property with a value less than the current size of the file shall truncate the file at the specified position. Writing a File_Size of 0 shall delete all of the file data but not the File object itself. Writing to the File_Size property with a value greater than the current size of the file shall expand the size of the file but the value of the new octets of the file shall be a local matter.

Devices may restrict the allowed values for writes to the File_Size. Specifically, devices may allow deletion of the file contents by writing a value of zero, but not necessarily allow arbitrary truncation or expansion.

If the size of the file is unknown, an attempt to read this property shall result in an Error Class of 'PROPERTY' and an Error Code of 'UNKNOWN_FILE_SIZE'.

[Add new Clause **18.3.X**, p. 392]

18.3.X UNKNOWN_FILE_SIZE - This error code is returned when the File_Size property is read and the size of the file is unknown.

[Change Clause **19.1.1**, p. 398]

19.1.1 The Backup and Restore Procedures

In BACnet building control systems, many devices will have configuration data that is set up by a vendor's proprietary configuration tool. This setup may consist of network visible BACnet objects and/or non-network visible settings. This section outlines the standard method that BACnet devices will employ if an interoperable device backup and restore feature is to be provided.

The backup and restore procedures use File objects to hold and transfer the configuration data. The content and format of the configuration files is a local matter. The choice of whether to use stream-based files or record-based files is a local matter. The services required to support the backup and restore procedures are ReinitializeDevice, ReadProperty, WriteProperty, AtomicWriteFile, AtomicReadFile, and, optionally, CreateObject, *ReadPropertyMultiple*, or *WritePropertyMultiple*.

[Change Clause 19.1.2.2, p. 398]

19.1.2.2 Preparation for Backup

Before starting a backup procedure, device A shall read the Backup_Preparation_Time property, if present, from device B's Device object. If the property is not present in device B, the value shall be assumed to be 0.

Upon receipt of the ReinitializeDevice(STARTBACKUP, <password>) message, if device B is able to perform a backup procedure, device B ~~will shall prepare for the backup procedure and~~ respond with a 'Result(+)' to the ReinitializeDevice service request. *Device B shall set its Backup_And_Restore_State to PREPARING_FOR_BACKUP. Upon receipt of a Result(+), device A shall monitor the Backup_And_Restore_State property and not continue with the backup until the property contains the value PERFORMING_A_BACKUP. During the time period immediately following the Result(+) defined by the Backup_Preparation_Time, device B is allowed to ignore requests from device A and as such device A shall not consider a lack of response during this period to be an error condition. It is a local matter whether device A initiates the monitoring of the Backup_And_Restore_State property during or after this time period. Once device B changes its Backup_And_Restore_State to PERFORMING_A_BACKUP, it shall not ignore requests from device A regardless of whether the Backup_Preparation_Time time period has expired.*

If device B is unable to perform a backup procedure or is already performing a backup procedure, then it ~~will shall~~ respond to the ReinitializeDevice service request with a 'Result(-)' response. Assuming device B supports the backup procedure and the request was properly formulated, the valid Error Class:Error Codes that can be returned are:

DEVICE:CONFIGURATION_IN_PROGRESS - if device B is already processing a backup or a restore request.

~~SERVICES:SERVICE_REQUEST_DENIED~~SECURITY:PASSWORD_FAILURE – if the password that was provided was incorrect or if a password is required and one was not provided.

After device B responds to the ReinitializeDevice request with a 'Result(+)', *device B has Backup_Preparation_Time seconds to prepare for the backup procedure. During this period of time, device B is not required to respond to any BACnet service requests. Once this period of time elapses, device B is required to respond to read requests for properties of the Device object. When device B has successfully completed its backup preparations in their entirety, the configuration File objects ~~must~~ shall exist in the ~~device~~ device and the Backup_And_Restore_State property shall be set to PERFORMING_A_BACKUP. The creation of configuration File objects during this time shall not have an effect on the Database_Revision property.*

If device B is unable to successfully complete its backup preparations, it shall set its Backup_And_Restore_State to BACKUP_FAILURE. Device A shall end the backup procedure when it detects device B's state is set to BACKUP_FAILURE.

It is a local matter as to whether device B will respond to other requests while ~~performing a backup procedure~~ ~~it is in backup mode~~. The exception to this is that device B ~~must~~ *is required to* accept and fulfill read requests by device A that consist of accesses to device B's Device object and/or its configuration File objects. *Note that Device B is allowed to return an UNKNOWN_FILE_SIZE error in response to requests for the File_Size property of any of its configuration files if the file size is unknown.* Any services that are rejected due to an in-progress backup procedure will be rejected with an error class of DEVICE and error code of DEVICE_BUSY.

It is a local matter as to whether device B will continue to perform control actions while it is in backup mode. If device B changes its operational behavior during a backup procedure, then the System_Status property of the Device object shall be set to BACKUP_IN_PROGRESS.

[Change Clause 19.1.2.5, p. 399]

19.1.2.5 Ending the Backup Procedure

When all of the configuration files have been read, device A sends a ReinitializeDevice(ENDBACKUP, <password>) message to device B. Device B will perform whatever actions are required to complete the backup in order to place the

device back into the state it was in before the backup procedure or into any other state as defined by the vendor. Device B must not remain in the BACKUP_IN_PROGRESS mode after the backup procedure has ended.

If device A needs to abort the backup for any reason (i.e., the user aborts the procedure, device B fails to allow reads from a configuration file, or device A detects any other condition that inhibits the backup procedure), device A shall attempt to send ReinitializeDevice(ENDBACKUP, <password>) to device B. Upon receipt of this message, device B shall end the backup procedure. If the backup procedure is aborted, device A should not assume that the configuration files are still valid and continue to read them.

The receipt of the ReinitializeDevice(ENDBACKUP, <password>) message shall cause device B to exit backup mode.

If device B does not receive any messages related to the backup procedure from device A for the number of seconds specified in the Backup_Failure_Timeout property of its Device object, device B should assume that the backup procedure has been aborted, and device B should exit backup mode. A message related to the backup procedure is defined to be any ReadProperty, *ReadPropertyMultiple*, WriteProperty, *WritePropertyMultiple*, CreateObject, or AtomicReadFile request that directly accesses a configuration File object.

When the backup procedure ends, device B shall set its Backup_and_Restore_State to IDLE. The deletion of configuration File objects during the backup procedure shall not have an effect on the Database_Revision property.

[Change Clause 19.1.3.2, p. 399]

19.1.3.2 Preparation for Restore

Before starting a restore procedure, device A shall read the Restore_Preparation_Time property from device B's Device object. If the property is not present in device B, the value shall be assumed to be 0.

Upon receipt of ~~the a~~ restore request, if device B is able to perform a restore procedure, device B ~~will prepare for the restore procedure and will~~ shall respond with a 'Result(+)' to the ReinitializeDevice service request. *Device B shall set its Backup_And_Restore_State to PREPARING_FOR_RESTORE.*

If device B is unable to perform a restore procedure, then it will respond to the ReinitializeDevice service request with a 'Result(-)' response. Assuming device B supports the restore procedure and the request was properly formulated, the valid Error Class:Error Codes that can be returned are:

DEVICE:CONFIGURATION_IN_PROGRESS – if device B is already processing a backup or a restore request.

~~SERVICES:SERVICE_REQUEST_DENIED SECURITY:PASSWORD_FAILURE~~ – if the password that was provided was incorrect or if a password is required and one was not provided.

After device B responds to the ReinitializeDevice request with a 'Result(+)', device B has Restore_Preparation_Time seconds to prepare for the restore procedure. During this period of time, device B is not required to respond to any BACnet service requests. Once this period of time elapses, device B is required to respond to read requests for properties of the Device object. When device B has completed its restore preparations in their entirety, the configuration File objects ~~must~~ shall exist in the device, or device B ~~must~~ shall be able to accept CreateObject requests from device A to create the configuration File objects, and the Backup_And_Restore_State property shall be set to PERFORMING_A_RESTORE. Once device B changes its Backup_And_Restore_State to PERFORMING_A_RESTORE, it shall not ignore requests from device A regardless of whether the Restore_Preparation_Time time period has expired. The creation of configuration File objects during the Restore procedure, whether automatically created by the device or by the execution of the CreateObject service, shall not impact the value of the Database_Revision property.

If device B is unable to successfully complete its restore preparations, it shall set its Backup_And_Restore_State to RESTORE_FAILURE. Device A shall abort the restore procedure when it detects device B's state is set to RESTORE_FAILURE.

Upon receipt of a Result(+), device A shall monitor the Backup_And_Restore_State property and not continue with the restore until the property contains the value PERFORMING_A_RESTORE. During the time period immediately following the Result(+) defined by the Restore_Preparation_Time, device B is allowed to ignore requests from device A and as such device A shall not consider a lack of response during this period to be an error condition. It is a local matter whether device A initiates the monitoring of the Backup_And_Restore_State property during or after this time period.

It is a local matter as to whether device B will respond to other requests while it is in restore mode. The exception to this is that device B must accept and fulfill read and write requests by device A that consist of accesses to device B's Device object and/or its configuration File objects. Any services that are rejected due to an in-progress backup procedure will be rejected with an error class of DEVICE and error code of CONFIGURATION_IN_PROGRESS.

Device B ~~must~~ shall be prepared to answer device A's requests for information from device B's Device object. If device B cannot service requests from devices other than device A, then device B shall reject those services with an error class of DEVICE and an error code of CONFIGURATION_IN_PROGRESS.

It is a local matter as to whether device B will continue to perform control actions while it is in restore mode. If device B changes its operational behavior during a restore procedure, then the System_Status property of the Device object shall be set to DOWNLOAD_IN_PROGRESS.

[Change Clause 19.1.3.4, p. 400]

19.1.3.4 Ending the Restore Procedure

When device A has completely written all of the configuration files to device B, device A ~~will~~ shall send ReinitializeDevice(ENDRESTORE, <password>). Device B will perform whatever actions are required to complete the restore procedure *within Restore_Completion_Time seconds after responding with a Result(+)*, which should include a validation of the restored configuration. If the validation fails, it is a local matter as to what device B will do beyond changing its System_Status property to something other than DOWNLOAD_IN_PROGRESS.

If device A needs to abort the restore for any reason (i.e., the user aborts the procedure, device B fails to allow writes to a configuration file, or device A detects any other condition that inhibits the restore procedure), device A shall attempt to send ReinitializeDevice(ABORTRESTORE, <password>) to device B. Upon receipt of this message, device B shall abort the restore procedure *within Restore_Completion_Time seconds after responding with a Result(+)*.

If device B does not receive any messages related to the restore procedure from device A for the number of seconds specified in the Backup_Failure_Timeout property of its Device object, device B should assume that the restore procedure has been aborted, and device B should exit restore mode. A message related to the restore procedure is defined to be any ReadProperty, ReadPropertyMultiple, WriteProperty, WritePropertyMultiple, CreateObject, or AtomicWriteFile request that directly accesses a configuration File object.

When the restore procedure ends successfully, device B shall set its Backup_and_Restore_State to IDLE and shall set the value of the Database_Revision property to the value it had before the restore, and then increment it.

...

[Change Clause 21, Error production, p. 445]

```

Error ::= SEQUENCE {
...
    timeout                               (30),
    unknown-file-size                     (122),
    unknown-object                        (31),
    ...
    -- see unknown-file-size              (122),
    ...

```



```

    }
    ...
}

```

[Add to Clause 21, new **BACnetBackupState** production, p. 449]

```

BACnetBackupState ::= ENUMERATED {
    idle (0),
    preparing-for-backup (1),
    preparing-for-restore (2),
    performing-a-backup (3),
    performing-a-restore (4),
    backup-failure (5),
    restore-failure (6)
}

```

[Change Clause 21, **BACnetPropertyStates** production, p. 471]

```

BACnetPropertyStates ::= CHOICE {
    ...
    door-alarm-state [15] BACnetDoorAlarmState,
    backup-state [36] BACnetBackupState
    ...
}

```

[Change Clause 21, **BACnetPropertyIdentifier** production, p.465]

```

BACnetPropertyIdentifier ::= ENUMERATED {
    ...
    average-value (125),
    backup-and-restore-state (338),
    backup-failure-timeout (153),
    backup-preparation-time (339),
    bias (14),
    ...
    resolution (106),
    restore-completion-time (340),
    restore-preparation-time (341),
    scale (187),
    ...
    -- see backup-and-restore-state (338),
    -- see backup-preparation-time (339)
    -- see restore-completion-time (340),
    -- see restore-preparation-time (341),
    ...
}

```

[Change Annex C, p. 501.]

```

...
DEVICE ::= SEQUENCE {
    ...
    backup-failure-timeout [153] Unsigned16,
    backup-preparation-time [339] Unsigned16,
    restore-preparation-time [341] Unsigned16,

```

```
restore-completion-time [340] Unsigned16,  
backup-and-restore-state [338] BACnetBackupState,  
active-cov-subscriptions [152] SEQUENCE OF BACnetCOVSubscription,  
...  
}
```

[Change Annex D.11, p. 512]

...

D.11 Examples of a Device Object

Example 1: A "sophisticated" BACnet device.

...

```
Property: Backup_Failure_Timeout = 300  
Property: Backup_Preparation_Time = 60  
Property: Restore_Preparation_Time = 120  
Property: Restore_Completion_Time = 240  
Property: Backup_And_Restore_State = IDLE  
Property: Active_COV_Subscriptions = (((0, (Device, Instance 12)), 300),  
((Analog Input, Instance 1),Present_Value),TRUE,100,1.0),  
(((0, (Device, Instance 40)), 600),  
((Analog Input, Instance 1),Present_Value),TRUE,3,1.5))
```

...

**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.