

BSR/ASHRAE Addendum *n*
to ANSI/ASHRAE Standard 135-2008

Public Review Draft

ASHRAE® Standard

Proposed Addendum *n* to Standard 135-2008, *BACnet*®—A *Data Communication Protocol for Building Automation and Control Networks*

Third Public Review (**March 2009**)
(Draft Shows Proposed Changes
to Current Standard)

This draft has been recommended for public review by the responsible project committee. To submit a comment on this proposed addendum, go to the ASHRAE website at <http://www.ashrae.org/technology/page/331> and access the online comment database. The draft is subject to modification until it is approved for publication by the Board of Directors and ANSI. 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 1-800-527-4723 (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 20, 2009**. 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



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

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

In the following document, language to be 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 proposed to be added, plain type is used throughout. Only this new and deleted text is open to comment as this time. All other material in this addendum is provided for context only and is not open for public review comment except as it relates to the proposed changes.

135-2008*n*-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 135-2008*n*-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 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 ReadProperty and ReadPropertyMultiple 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 ReadProperty and ReadPropertyMultiple 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      (n),  
    unknown-object         (31),  
    ...  
    -- see unknown-file-size (n),  
    ...
```

```

    }
    ...
}

```

[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 [x] BACnetBackupState
    ...
}

```

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

```

BACnetPropertyIdentifier ::= ENUMERATED {
    ...
    average-value (125),
    -- see backup-and-restore-state (n),
    backup-failure-timeout (153),
    -- see backup-preparation-time (n+1),
    bias (14),
    ...
    resolution (106),
    -- see restore-completion-time (n+2),
    -- see restore-preparation-time (n+3),
    scale (187),
    ...
    backup-and-restore-state (n),
    backup-preparation-time (n+1),
    restore-completion-time (n+2),
    restore-preparation-time (n+3),
    ...
}

```

[Change Annex C, p. 501.]

```

...
DEVICE ::= SEQUENCE {
    ...
    backup-failure-timeout [153] Unsigned16,
    backup-preparation-time [n+1] Unsigned16,
    restore-preparation-time [n+3] Unsigned16,

```

```
restore-completion-time [n+2] Unsigned16,  
backup-and-restore-state [n] 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))
```

...