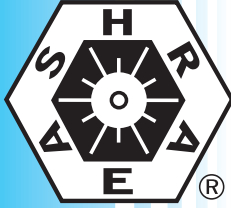


ANSI/ASHRAE Addendum f to
ANSI/ASHRAE Standard 135-2004



ASHRAE STANDARD



BACnet[®]

A Data Communication Protocol for Building Automation and Control Networks

Approved by the ASHRAE Standards Committee on January 27, 2007; by the ASHRAE Board of Directors on March 25, 2007; and by the American National Standards Institute on March 26, 2007.

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 2007 ASHRAE, Inc.

ISSN 1041-2336



www.ansi.org

**American Society of Heating, Refrigerating
and Air-Conditioning Engineers, Inc.**

1791 Tullie Circle NE, Atlanta, GA 30329

www.ashrae.org

ASHRAE Standing Standard Project Committee 135
Cognizant TC: TC 1.4, Control Theory and Application
SPLS Liaison: Frank E. Jakob

William O. Swan, III, <i>Chair*</i>	Steven T. Bushby	J. Damian Ljungquist*
David Robin, <i>Vice-Chair</i>	A. J. Capowski	James G. Luth*
Carl Neilson, <i>Secretary</i>	Keith A. Corbett	Jerald P. Martocci
Donald P. Alexander*	Craig P. Gemmill	Carl J. Ruther*
Barry B. Bridges*	David G. Holmberg*	David B. Thompson*
Coleman L. Brumley, Jr.*	Robert L. Johnson*	Daniel A. Traill
Ernest C. Bryant*	Stephen Karg	J. Michael Whitcomb*
James F. Butler*		David F. White

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

ASHRAE STANDARDS COMMITTEE 2006–2007

David E. Knebel, <i>Chair</i>	James D. Lutz
Stephen D. Kennedy, <i>Vice-Chair</i>	Carol E. Marriott
Michael F. Beda	Merle F. McBride
Donald L. Brandt	Mark P. Modera
Steven T. Bushby	Ross D. Montgomery
Paul W. Cabot	H. Michael Newman
Hugh F. Crowther	Stephen V. Santoro
Samuel D. Cummings, Jr.	Lawrence J. Schoen
Robert G. Doerr	Stephen V. Skalko
Roger L. Hedrick	Bodh R. Subherwal
John F. Hogan	Jerry W. White, Jr.
Eli P. Howard, III	James E. Woods
Frank E. Jakob	Richard D. Hermans, <i>BOD ExO</i>
Jay A. Kohler	Hugh D. McMillan, III, <i>CO</i>

Claire B. Ramspeck, *Assistant Director of Technology for Standards and Special Projects*

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,
- 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.)

The purpose of this addendum is to revise ANSI/ASHRAE Standard 135-2004. The modifications in this addendum are the result of change proposals made pursuant to the ASHRAE continuous maintenance procedures and of deliberations within Standing Standard Project Committee 135.

SSPC 135 wishes to recognize the efforts of the following people in developing this addendum: Howard Coleman, Sharon Dinges, Stuart Donaldson, David Fisher, John Hartman, Bernhard Isler, Simon Lemaire, Les Mather, Kornelia Mergner, Hans-Joachim Mundt, Masahiro Ogawa, David Ritter, Stephen Treado, David White, and Rob Zivney.

The changes in Addendum 135f are summarized below.

135-2004f-1. Add new Access Door object type, p. 1.

In the following document, language to be added to existing clauses of ANSI/ASHRAE 135-2004 and Addenda is indicated through the use of *italics*, while deletions are indicated by ~~striethrough~~. Where entirely new clauses or subclauses are to be added, plain type is used throughout.

135-2004f-1. Add new Access Door object type.

Rationale

There is need for a standard BACnet object which represents the physical characteristics of an access-controlled door. This Access Door object represents all the physical door hardware parts that are commonly associated with a door such as a door contact, door lock, request-to-exit device, etc. Through this object the door may be commanded to be locked, unlocked or pulse-unlocked (unlocked for a specified period of time). In addition, this object may generate access control alarms for conditions such as door-open-too-long, forced-open etc.

Addendum 135-2004f-1

[Add new Clause **12.X**, p. 251]

12.X Access Door Object Type

The Access Door object type is an abstract interface to a physical door whose properties represent the externally visible characteristics of an access control door. The Access Door is comprised of a collection of physical door hardware, such as a door lock, a door contact, and a Request-To-Exit device, which together comprise a door for access control. The individual hardware components of the door may or may not be exposed through this object. The object and its properties are summarized in Table 12-X1 and described in detail in this subclause.

Table 12-X1. Properties of the Access Door Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	BACnetDoorValue	W
Description	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	R
Out_Of_Service	BOOLEAN	R
Priority_Array	BACnetPriorityArray	R
Relinquish_Default	BACnetDoorValue	R
Door_Status	BACnetDoorStatus	O ^{1,2}
Lock_Status	BACnetLockStatus	O ¹
Secured_Status	BACnetDoorSecuredStatus	O
Door_Members	BACnetARRAY[N] of BACnetDeviceObjectReference	O
Door_Pulse_Time	Unsigned	R
Door_Extended_Pulse_Time	Unsigned	R
Door_Unlock_Delay_Time	Unsigned	O
Door_Open_Too_Long_Time	Unsigned	R
Door_Alarm_State	BACnetDoorAlarmState	O ^{1,3}
Masked_Alarm_Values	List of BACnetDoorAlarmState	O
Maintenance_Required	BACnetMaintenance	O
Time_Delay	Unsigned	O ³
Notification_Class	Unsigned	O ³
Alarm_Values	List of BACnetDoorAlarmState	O ³
Fault_Values	List of BACnetDoorAlarmState	O ³
Event_Enable	BACnetEventTransitionBits	O ³
Acked_Transitions	BACnetEventTransitionBits	O ³
Notify_Type	BACnetNotifyType	O ³
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	O ³
Profile_Name	CharacterString	O

¹ These properties, when present, shall be writable when Out_Of_Service is TRUE.

² This property shall be required if the property Secured_Status is present.

³ These properties are required if the object supports intrinsic reporting.

12.X.1 Object_Identifier

This property, of type BACnetObjectIdentifier, is a numeric code that is used to identify the object. It shall be unique within the BACnet Device that maintains it.

12.X.2 Object_Name

This property, of type CharacterString, shall represent a name for the object that is unique within the BACnet Device that maintains it. The minimum length of the string shall be one character. The set of characters used in the Object_Name shall be restricted to printable characters.

12.X.3 Object_Type

This property, of type BACnetObjectType, indicates membership in a particular object-type class. The value of this property shall be ACCESS_DOOR.

12.X.4 Present_Value (Commandable)

This property, of type BACnetDoorValue, reflects the current active command of the access door object. The Present_Value is commandable and has one of the following values:

LOCK	The door is commanded to the locked state.
UNLOCK	The door is commanded to the unlocked state.
PULSE_UNLOCK	The door will be commanded to the unlocked state for a maximum of the time specified by Door_Pulse_Time, after which the value will be automatically relinquished from the priority array at the commanded priority. It is permissible for the local controller to relinquish the value from the priority array before the time specified by Door_Pulse_Time has expired. The conditions when this may occur are considered a local matter.

If a value of PULSE_UNLOCK is written at a given priority and the Present_Value is currently being commanded, at any value, at a higher priority then the lower priority value will be relinquished immediately.

EXTENDED_PULSE_UNLOCK	The door will be commanded to the unlocked state for a maximum of the time specified by Door_Extended_Pulse_Time, after which the value will be automatically relinquished from the priority array at the commanded priority. It is permissible for the local controller to relinquish the value from the priority array before the time specified by Door_Extended_Pulse_Time has expired. The conditions when this may occur are considered a local matter.
-----------------------	---

If a value of EXTENDED_PULSE_UNLOCK is written at a given priority and the Present_Value is currently being commanded, at any value, at a higher priority then the lower priority value will be relinquished immediately.

Note that the present value represents the commanded state of the door, which does not necessarily correspond to the physical state of the door lock.

The present value of the Access Door is defined for a standard access controlled door, where the control operation is to lock or unlock. However, this does not exclude motorized devices such as sliding doors, parking gates, etc where the operation is to open or close. In these cases, locked shall be equivalent to closed and unlocked shall be equivalent to open.

12.X.5 Description

This property, of type CharacterString, is a string of printable characters whose content is not restricted.

12.X.6 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of the physical door. Three of the flags are associated with the values of other properties of this object. A more detailed status could be determined by reading the properties that are linked to these flags. The relationship between individual flags is not defined by the protocol. The four flags are:

{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}

Where:

IN_ALARM	Logical FALSE (0) if the Event_State property has a value of NORMAL, otherwise logical TRUE (1).
FAULT	Logical TRUE (1) if the Reliability property is present and does not have a value of NO_FAULT_DETECTED, otherwise logical FALSE (0).
OVERRIDDEN	Logical TRUE (1) if the object has been overridden by some mechanism local to the BACnet Device. In this context “overridden” is taken to mean that the physical door is no longer tracking changes to the Present_Value property and the Reliability property is no longer a reflection of the reliability of the physical inputs(s) and output(s). Otherwise, the value is logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property has a value of TRUE, otherwise logical FALSE (0).

12.X.7 Event_State

The Event_State property, of type BACnetEventState, is included in order to provide a way to determine if this object has an active event state associated with it. The Event_State property shall indicate the event state of the object. If the object does not support intrinsic reporting, then the value of this property shall be NORMAL.

12.X.8 Reliability

The Reliability property, of type BACnetReliability, provides an indication of whether the Present_Value or the operation of the physical inputs or outputs which comprise this door are "reliable" as far as the BACnet Device or operator can determine and, if not, why. The Reliability property for this object may have any of the following values:

{NO_FAULT_DETECTED, MULTISTATE_FAULT, CONFIGURATION_ERROR, UNRELIABLE_OTHER}.

12.1.8.1 Conditions for Generating a TO-FAULT Event

A TO-FAULT event is generated under these conditions:

- (a) the Reliability property becomes not equal to NO_FAULT_DETECTED, and
- (b) the TO-FAULT flag must be enabled in the Event_Enable property.

12.X.9 Out_Of_Service

The Out_Of_Service property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the logical door which this object represents is not in service. This means that the Present_Value property is decoupled from the physical door and will not track changes to the physical door when the value of Out_Of_Service is TRUE. In addition, the Reliability property and the corresponding state of the FAULT flag of the Status_Flags property shall be decoupled from the physical door when Out_Of_Service is TRUE. While the Out_Of_Service property is TRUE, the Present_Value and Reliability properties, and if present, the Door_Status, Lock_Status and Door_Alarm_State properties, may be changed to any value as a means of simulating specific fixed conditions or for testing purposes. Other functions that depend on the state of the Present_Value or Reliability properties, and if present the Door_Status, Lock_Status and Door_Alarm_State properties, shall respond to changes made to these properties while Out_Of_Service is TRUE, as if those changes had occurred to the physical door.

12.X.10 Priority_Array

This property is a read-only array that contains prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism.

12.X.11 Relinquish_Default

This property is the default value to be used for the Present_Value property when all command priority values in the Priority_Array property have a NULL value. See Clause 19. The acceptable values for this property are either LOCK or UNLOCK and the property shall not take on either of the values PULSE_UNLOCK or EXTENDED_PULSE_UNLOCK.

12.X.12 Door_Status

This optional property, of type BACnetDoorStatus, represents the open or closed state of the physical door. The values that may be taken on by this property are:

CLOSED	The door is closed.
OPENED	The door is open or partially open.
UNKNOWN	It is unknown whether the door is opened or closed.

This property, if present, is required to be writable when Out_Of_Service is TRUE.

12.X.13 Lock_Status

This optional property, of type BACnetLockStatus, represents the monitored (as opposed to the commanded) status of the door lock. The values that may be taken on by this property are:

LOCKED	The door lock is locked.
UNLOCKED	The door lock is unlocked.
FAULT	The lock status input associated with the door lock is unreliable.
UNKNOWN	There is no lock status input associated with the door and therefore the status of the physical lock is unknown.

This property, if present, is required to be writable when Out_Of_Service is TRUE.

12.X.14 Secured_Status

This optional property, of type BACnetDoorSecuredStatus, represents whether or not the physical door is in a secured state. This property shall have a value of SECURED if, and only if, all of the following conditions are met:

- (a) the IN_ALARM flag of the Status_Flags property is FALSE, and
- (b) the Masked_Alarm_Values list, if it exists, is empty, and
- (c) the Door_Status property has a value of CLOSED, and
- (d) the Present_Value property has a value of LOCK, and
- (e) the Lock_Status property, if it exists, has a value of LOCKED or UNKNOWN.

If one or more of the previous conditions are not met, the property shall have a value of UNSECURED. If the device can not determine any of the previous conditions, then the property shall have a value of UNKNOWN.

12.X.15 Door_Members

This optional property, of type BACnetARRAY[N] of BACnetDeviceObjectReference holds an array of references to BACnet objects which represent I/O devices, authentication devices, schedules, programs, or other objects that are associated with the physical door. It is a local matter as to how this array is used and which objects are referenced in this array. The array may be empty or not present if the vendor does not wish to expose the individual objects that make up this physical door.

12.X.16 Door_Pulse_Time

This property, of type Unsigned, is the maximum duration of time, in tenths of seconds, for which the door will be unlocked when the Present_Value has a value of PULSE_UNLOCK, after which time the Present_Value shall be automatically relinquished at the priority that established the PULSE_UNLOCK command.

12.X.17 Door_Extended_Pulse_Time

This property, of type Unsigned, is the maximum amount of time, in tenths of seconds, which the door will be unlocked when the Present_Value has a value of EXTENDED_PULSE_UNLOCK, after which time the Present_Value shall be automatically relinquished at the priority that established the EXTENDED_PULSE_UNLOCK command.

12.X.18 Door_Unlock_Delay_Time

This optional property, of type Unsigned, is the duration of time, in tenths of seconds, which the physical door lock will delay unlocking when the Present_Value changes to a value of PULSE_UNLOCK or EXTENDED_PULSE_UNLOCK.

12.X.19 Door_Open_Too_Long_Time

This property, of type Unsigned, is the time, in tenths of seconds, to delay before setting the Door_Alarm_State to DOOR_OPEN_TOO_LONG after it is determined that a door-open-too-long condition exists. A door-open-too-long condition occurs when the Present_Value has a value of LOCK and one of the following conditions exist:

- (a) The Present_Value had a previous value of PULSE_UNLOCK and the door has been in a continual open state for the time specified by Door_Open_Too_Long_Time after the Door_Pulse_Time has expired.
- (b) The Present_Value had a previous value of EXTENDED_PULSE_UNLOCK and the door has been in a continual open state for the time specified by Door_Open_Too_Long_Time after the Door_Extended_Pulse_Time has expired.
- (c) The Present_Value had a previous value of UNLOCK and the door has been in a continual open state for the time specified by Door_Open_Too_Long_Time.

12.X.20 Door_Alarm_State

This optional property, of type BACnetDoorAlarmState, is the alarm state for the physical door. The CHANGE_OF_STATE event algorithm will monitor this property. When no alarm or fault condition exists for this object, this property shall take on the value NORMAL. It is considered a local matter as to when this property is set to a non-normal value. It is up to the internal control logic to take Lock_Status, Door_Status, Present_Value and information from other objects into account when calculating the proper alarm state. However, this property cannot take on any value which is also in the Masked_Alarm_Values list. If the property is currently set to a specific state and that state is written to the Masked_Alarm_Values list, then the Door_Alarm_State will immediately return to the NORMAL state.

This property is required if intrinsic reporting is supported by this object, and if present, required to be writable when Out_Of_Service is TRUE.

12.X.21 Masked_Alarm_Values

This optional property, of type List of BACnetDoorAlarmState, shall specify any alarm and/or fault states which are masked. An alarm state which is currently masked will prevent the Door_Alarm_State property from being equal to that state.

12.X.22 Maintenance_Required

This optional property, of type BACnetMaintenance, shall indicate the type of maintenance required for the Access Door. This may be periodic maintenance, or a "parameter-determined" maintenance, such as maximum duty-cycle for a door lock, and shall be determined locally.

12.X.23 Time_Delay

This optional property, of type Unsigned, shall specify the minimum period of time in seconds that the Door_Alarm_State must remain equal to any one of the values in the Alarm_Values property before a TO-OFFNORMAL event is generated or remain not equal to any of the values in the Alarm_Values property before a TO-NORMAL event is generated. This property is required if intrinsic reporting is supported by this object.

12.X.24 Notification_Class

This optional property, of type Unsigned, shall specify the notification class to be used when handling and generating event notifications for this object. The Notification_Class property implicitly refers to a Notification Class object that has a Notification_Class property with the same value. This property is required if intrinsic reporting is supported by this object.

12.X.25 Alarm_Values

This optional property, of type List of BACnetDoorAlarmState, shall specify any states the Door_Alarm_State must equal before a TO-OFFNORMAL event is generated and event state OFFNORMAL is entered. This property is required if intrinsic reporting is supported by this object.

12.X.25.1 Conditions for Generating a TO-OFFNORMAL Event

A TO-OFFNORMAL event is generated under these conditions:

- (a) the TO-OFFNORMAL flag must be enabled in the Event_Enable property, and
- (b) the Door_Alarm_State must equal any of the values in the Alarm_Values list, and
- (c) the Door_Alarm_State must remain equal to the same value within the Alarm_Values list for a minimum period of time specified by the Time_Delay property.

12.X.25.2 Conditions for Generating a TO-NORMAL Event

Once equal, the Door_Alarm_State must become not equal to any of the states in the Alarm_Values property, and not equal to any of the states in the Fault_Values property, before a TO-NORMAL event is generated under these conditions:

- (a) the TO-NORMAL flag must be enabled in the Event_Enable property, and
- (b) the Door_Alarm_State must remain not equal to any of the states in the Alarm_Values property, and
- (c) the Door_Alarm_State must remain not equal to any of the states in the Fault_Values property, and
- (d) the Door_Alarm_State must remain equal to the same value for a minimum period of time, specified by the Time_Delay property.

12.X.26 Fault_Values

This optional property, of type List of BACnetDoorAlarmState, shall specify any states the Door_Alarm_State must equal before a TO-FAULT event is generated. If Door_Alarm_State becomes equal to any of the states in the Fault_Values list and no physical fault has been detected, then the Reliability property shall have the value MULTI_STATE_FAULT. This property is required if intrinsic reporting is supported by this object.

12.X.26.1 Conditions for Generating a TO-FAULT Event

A TO-FAULT event is generated under these conditions:

- (a) the TO-FAULT flag must be enabled in the Event_Enable property, and
- (b) the Door_Alarm_State must equal any of the values in the Fault_Values property.

12.X.26.2 Conditions for Generating a TO-NORMAL Event

Conditions for generating a TO-NORMAL event are defined in 12.X.25.2.

12.X.27 Event_Enable

This property, of type BACnetEventTransitionBits, shall convey three flags that separately enable and disable reporting of TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. This property is required if intrinsic reporting is supported by this object.

12.X.28 Acked_Transitions

This optional property, of type BACnetEventTransitionBits, shall convey three flags that separately indicate the receipt of acknowledgments for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events. These flags shall be cleared upon the occurrence of the corresponding event and set under any of these conditions:

- (a) upon receipt of the corresponding acknowledgment;
- (b) upon the occurrence of the event if the corresponding flag is not set in the Event_Enable property (meaning event notifications will not be generated for this condition and thus no acknowledgment is expected);
- (c) upon the occurrence of the event if the corresponding flag is set in the Event_Enable property and the corresponding flag in the Ack_Required property of the Notification Class object implicitly referenced by the Notification_Class property of this object is not set (meaning no acknowledgment is expected).

This property is required if intrinsic reporting is supported by this object.

12.X.29 Notify_Type

This optional property, of type BACnetNotifyType, shall convey whether the notifications generated by the object should be Events or Alarms. This property is required if intrinsic reporting is supported by this object.

12.X.30 Event_Time_Stamps

This optional property, of type BACnetARRAY[3] of BACnetTimeStamp, shall convey the times of the last event notifications for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events, respectively. Time stamps of type Time or Date shall have X'FF' in each octet and Sequence number time stamps shall have the value 0 if no event notification of that type has been generated since the object was created. This property is required if intrinsic reporting is supported by this object.

12.X.31 Profile_Name

This optional property, of type `CharacterString`, is the name of an object profile to which this object conforms. To ensure uniqueness, a profile name must begin with a vendor identifier code (see Clause 23) in base-10 integer format, followed by a dash. All subsequent characters are administered by the organization registered with that vendor identifier code. The vendor identifier code that prefixes the profile name shall indicate the organization that publishes and maintains the profile document named by the remainder of the profile name. This vendor identifier need not have any relationship to the vendor identifier of the device within which the object resides.

A profile defines a set of additional properties, behavior, and/or requirements for this object beyond those specified here. This standard defines only the format of the names of profiles. The definition of the profiles themselves is outside the scope of this standard.

[Add the following entry to **Table 13.1**, p. 254]

Table 13-1. Standardized Objects That May Support COV Reporting

Object Type	Criteria	Properties Reported
<i>Access Door</i>	<i>If Present_Value changes at all or Status_Flags changes at all or Door_Alarm_State changes at all (if the object has a Door_Alarm_State property)</i>	<i>Present_Value, Status_Flags, Door_Alarm_State (if the object has a Door_Alarm_State property)</i>
...		

[Add the following entry to **Table 13.2**, p. 256]

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

Object Type	Criteria	Event Type
<i>Access Door</i>	<i>If Door_Alarm_State changes to a new state for longer than Time_Delay AND the new transition is enabled in Event_Enable</i>	<i>CHANGE_OF_STATE</i>
...

[Add the following entry to **Table 13.3**, p. 257]

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

Object	Event Type	Notification Parameters	Referenced Object's Properties
<i>Access Door</i>	<i>CHANGE_OF_STATE</i>	<i>New_State Status_Flags</i>	<i>Door_Alarm_State Status_Flags</i>
....

[Add the following entry to **19.2.1.1**, “Commandable Properties”, p. 363]

The prioritization scheme is applied to certain properties of objects. The standard commandable properties and objects are as follows:

<u>OBJECT</u>	<u>COMMANDABLE PROPERTY</u>
Analog Output	Present_Value
Binary Output	Present_Value
Multi-state Output	Present_Value
Multi-state Value	Present_Value
Analog Value	Present_Value
Binary Value	Present_Value
<i>Access Door</i>	<i>Present_Value</i>

The designated properties of the Analog Output, Binary Output, ~~and~~ Multi-state Output *and* Access Door objects are commandable (prioritized) by definition. The designated properties of the Analog Value, Binary Value and Multi-state Value objects may optionally be commandable. Individual vendors, however, may decide to apply prioritization to any of the vendor-specified properties. These additional commandable properties shall have associated Priority_Array and Relinquish_Default properties with appropriate names. See 23.3.

[Add the following enumerations to **Clause 21** “Formal Descriptions of Application Protocol Data Units”, p. 390]

```
BACnetDoorValue ::= ENUMERATED {  
    lock           (0),  
    unlock         (1),  
    pulse-unlock  (2),  
    extended-pulse-unlock (3)  
}
```

```
BACnetLockStatus ::= ENUMERATED {  
    locked        (0),  
    unlocked     (1),  
    fault        (2),  
    unknown      (3)  
}
```

```
BACnetDoorSecuredStatus ::= ENUMERATED {  
    secured       (0),  
    unsecured    (1),  
    unknown      (2)  
}
```

```
BACnetDoorStatus ::= ENUMERATED {  
    closed       (0),  
    opened      (1),  
    unknown     (2)  
}
```

```
BACnetDoorAlarmState ::= ENUMERATED {  
    normal       (0),  
    alarm       (1),  
    door-open-too-long (2),  
    forced-open (3),  
    tamper      (4),  
    door-fault  (5),  
    lock-down   (6),  
    free-access (7),  
    egress-open (8),  
    ...  
}
```

-- 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. The last enumeration used in this version is 8.

[Add the following to the **BACnetObjectType** production in **Clause 21**, p. 421]

```
BACnetObjectType ::= ENUMERATED {  
    access-door      (30),  
    accumulator     (23),  
    analog-input    (0),  
    analog-output   (1),  
    analog-value    (2),  
    ...  
    -- see access-door (30)  
}
```

[Add the following to the **BACnetObjectTypesSupported** production in **Clause 21**, p. 422]

```
BACnetObjectTypesSupported ::= BIT STRING {  
  -- access-door           (30),  
  -- accumulator           (23),  
  analog-input             (0),  
  ...  
  -- trend-log             (20),  
  -- Objects added after 1995  
  averaging                (18),  
  multi-state-value        (19),  
  trend-log                (20),  
  life-safety-point        (21),  
  life-safety-zone         (22),  
  -- Objects added after 2001  
  accumulator              (23),  
  pulse-converter          (24),  
  -- Objects added after 2004  
  access-door             (30)  
}
```

[Add the following to the **BACnetPropertyIdentifier** production in **Clause 21**, p. 423]

```
BACnetPropertyIdentifier ::= ENUMERATED {  
  ...  
  direct-reading           (156),  
  door-alarm-state         (226),  
  door-extended-pulse-time (227),  
  door-members           (228),  
  door-open-too-long-time (229),  
  door-pulse-time        (230),  
  door-status            (231),  
  door-unlock-delay-time (232),  
  effective-period         (32),  
  ...  
  location                 (58),  
  lock-status            (233),  
  log-buffer               (131),  
  ...  
  manual-slave-address-binding (170),  
  masked-alarm-values     (234),  
  maximum-output          (61),  
  ...  
  schedule-default        (174),  
  secured-status         (235),  
  segmentation-supported   (107),  
  ...  
  -- see value-change-time (192),  
  -- see door-alarm-state (226),  
  -- see door-extended-pulse-time (227),  
  -- see door-members (228),  
  -- see door-open-too-long-time (229),  
  -- see door-pulse-time (230),  
  -- see door-status (231),  
  -- see door-unlock-delay-time (232),  
  -- see lock-status (233),
```

```

-- see masked-alarm-values    (234),
-- see secured-status        (235),
...
}

```

```

-- 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.

```

[Add the following to the **BACnetPropertyStates** production in **Clause 21**, p. 428]

```

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,
    binary-value       [1] BACnetBinaryPV,
    event-type         [2] BACnetEventType,
    polarity           [3] BACnetPolarity,
    program-change     [4] BACnetProgramRequest,
    program-state      [5] BACnetProgramState,
    reason-for-halt    [6] BACnetProgramError,
    reliability        [7] BACnetReliability,
    state              [8] BACnetEventState,
    system-status      [9] BACnetDeviceStatus,
    units              [10] BACnetEngineeringUnits,
    unsigned-value     [11] Unsigned,
    life-safety-mode   [12] BACnetLifeSafetyMode,
    life-safety-state  [13] BACnetLifeSafetyState,
    --enumeration value 14 is used in proposed Addendum 135-2004b-5
    door-alarm-state   [15] BACnetDoorAlarmState
    ...
}

```

[Add the following entry to **Table 23.1**, p. 437]

Table 23-1. Extensible Enumerations

Enumeration Name	Reserved Range	Maximum Value
...		
BACnetDeviceStatus	0-63	65535
<i>BACnetDoorAlarmState</i>	0-255	65535
BACnetEngineeringUnits	0-255	65535
...		

[Add to **Annex C**, p. 453]

```
ACCESS-DOOR::= SEQUENCE {
    object-identifier      [75]  BACnetObjectIdentifier,
    object-name            [77]  CharacterString,
    object-type           [79]  BACnetObjectType,
    present-value         [85]  BACnetDoorValue,
    description           [28]  CharacterString OPTIONAL,
    status-flags          [111]  BACnetStatusFlags,
    event-state           [36]  BACnetEventState,
    reliability           [103]  BACnetReliability,
    out-of-service        [81]  BOOLEAN,
    priority-array        [87]  BACnetPriorityArray,
    relinquish-default    [104]  BACnetDoorValue,
    door-status           [231]  BACnetDoorStatus,
    lock-status           [233]  BACnetLockStatus,
    secured-status        [235]  BACnetDoorSecuredStatus,
    door-members          [228]  SEQUENCE OF BACnetDeviceObjectReference OPTIONAL,
                                --accessed as a BACnetARRAY

    door-pulse-time       [230]  Unsigned,
    door-extended-pulse-time [227] Unsigned OPTIONAL,
    door-unlock-delay-time [232] Unsigned OPTIONAL,
    door-open-too-long-time [229] Unsigned,
    door-alarm-state      [226] BACnetDoorAlarmState OPTIONAL,
    masked-alarm-values   [234] SEQUENCE OF BACnetDoorAlarmState OPTIONAL,
    maintenance-required [158] BACnetMaintenance OPTIONAL,
    time-delay            [113] Unsigned OPTIONAL,
    notification-class    [17]  Unsigned OPTIONAL,
    alarm-values          [7]   SEQUENCE OF BACnetDoorAlarmState OPTIONAL,
    fault-values          [39]  SEQUENCE OF BACnetDoorAlarmState OPTIONAL,
    event-enable          [35]  BACnetEventTransitionBits OPTIONAL,
    acked-transitions     [0]   BACnetEventTransitionBits OPTIONAL,
    notify-type           [72]  BACnetNotifyType OPTIONAL,
    event-time-stamps     [130] SEQUENCE OF BACnetTimeStamp OPTIONAL,
                                --accessed as a BACnetARRAY

    profile-name          [168]  CharacterString OPTIONAL
}
```

[Add the following new clause to **Annex D**, p. 465, and renumber subsequent clauses]

D.1 Example of an Access Door object

This example of an access door represents the main entrance doors to an office building. The door is scheduled for UNLOCK at priority 10 but this has been overridden with a LOCK by the manual operator at priority 8. The door is currently locked but is opened which has resulted in a forced-open alarm condition. The door-open-too-long alarm is temporarily masked and will not generate this type of alarm regardless of how long the door is held open.

```
Property:  object-identifier =      (Access Door, Instance 1)
Property:  object-name =           "Main Entrance"
Property:  object-type =           ACCESS DOOR
Property:  present-value =         LOCK
Property:  description =           "Main entrance south building"
Property:  status-flags =          (TRUE, FALSE, TRUE, FALSE)
Property:  event-state =           OFFNORMAL
Property:  reliability =           NO_FAULT_DETECTED
Property:  out-of-service =        FALSE
Property:  priority-array =        {NULL, NULL, NULL, NULL, NULL, NULL, NULL, LOCK, NULL,
                                   UNLOCK, NULL, NULL, NULL, NULL, NULL, NULL}

Property:  relinquish-default =    LOCK
Property:  door-status =           OPEN
Property:  lock-status =           LOCKED
Property:  secured-status =        UNSECURED
Property:  door-members =          {(Multistate Input, Instance 1),
                                   (Binary Output, Instance 7),
                                   (Binary Input, Instance 1),
                                   (Schedule, Instance 4)}

Property:  door-pulse-time =       60
Property:  door-extended-pulse-time = 300
Property:  door-open-too-long-time = 300
Property:  door-alarm-state =      FORCED_OPEN
Property:  masked-alarm-values =   {DOOR_OPEN_TOO_LONG}
Property:  maintenance-required =  NONE
Property:  time-delay =            0
Property:  notification-class =    1
Property:  alarm-values =          {DOOR_OPEN_TOO_LONG, FORCED_OPEN, TAMPER}
Property:  fault-values =          {DOOR_FAULT}
Property:  event-enable =          {TRUE, FALSE, TRUE}
Property:  acked-transitions =     {FALSE, TRUE, TRUE}
Property:  notify-type =           ALARM
Property:  event-time-stamps =     ((7-JAN-2006, 15:30:51.7),
                                   (*-*,*:*:*)),
                                   (*-*,*:*:*))
```

[Add a new entry to **History of Revisions**, p.598]

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

HISTORY OF REVISIONS

<i>Protocol</i>		<i>Summary of Changes to the Standard</i>
<i>Version</i>	<i>Revision</i>	
...
1	6	Addendum f to ANSI/ASHRAE 135-2004 Approved by the ASHRAE Standards Committee January 27, 2007; by the ASHRAE Board of Directors March 25, 2007; and by the American National Standards Institute March 26, 2007. 1. Add new Access Door object type.

**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.