BSR/ASHRAE Addendum *e*
to ANSI/ASHRAE Standard 135-2004

Public Review Draft

ASHRAE® Standard

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

First Public Review (September 2005)
(Complete Draft for Full Review)

This draft has been recommended for public review by the responsible project committee. To submit a comment on this proposed addendum, use the comment form and instructions provided with this draft. The draft is subject to modification until it is approved for publication by the responsible project committee, the ASHRAE Standards Committee, and the Board of Directors. Then it will be submitted to the American National Standards Institute Board of Standards Review (BSR) for approval. Until this time, the current edition of the standard (as modified by any published addenda on the ASHRAE web site) remains in effect. The current edition of any standard may be purchased from the ASHRAE Bookstore @ http://www/ashrae.org or by calling 404-636-8400 or 1-800-527-4723 (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.

© August 11, 2005. This draft is covered under ASHRAE copyright. Permission to reproduce or redistribute all or any part of this document must be obtained from the ASHRAE Manager of Standards, 1791 Tullie Circle, NE, Atlanta, GA 30329-2305. Phone: 404-636-8400, Ext1125. Fax: 404-321-5478. E-mail: standards.section@ashrae.org.

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

[This foreword and the "rationale" on the following page are not part of this standard. They are merely informative and do not contain requirements necessary for conformance to the standard.]

FOREWORD

The purpose of this 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.

135-2004e-1. Add a new Load Control 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 ~~strikethrough~~. Where entirely new subclauses are proposed to be added, plain type is used throughout. Only this new and deleted text is open to comment at this time. All other material in this addendum is provided for context only and is not open to public review comment except as it relates to the proposed changes.

135-2004*e*-1. Add a new Load Control object type.

Rationale
There is need for a standard object type to allow a standard means for providing external control over load shedding.

Addendum 135-2004*e*-1

[Add new **Clause 12.17**, p. 206, and renumber existing **Clause 12.17** and subsequent clauses, including tables and figures]

12.17 Load Control Object Type

The Load Control object type defines a standardized object whose properties represent the externally visible characteristics of a mechanism for controlling load requirements. A BACnet device can use a Load Control object to allow external control over the shedding of a load that it controls. The mechanisms by which the loads are shed are not visible to the BACnet client. One or more objects may be used in the device to allow independent control over different sub-loads. The Load Control object may also be used in a hierarchical fashion to control other Load Control objects in other BACnet devices.

A BACnet client (controller) can request that the Load Control object shed a portion of its load for a specified time by writing to the four properties: Requested_Shed_Level, Start_Time, Shed_Duration, and Duty_Window. For any given shed request, which may arrive while a previous request is pending or active, each of these parameters is optional except for Start_Time, which must be written if no shed request is pending or active. If no shed request is pending or active, only the writing of Start_Time will cause the Load Control object to become active. Modification of these shed request parameters serves to configure the load shed command. Initial values of these properties, and the values taken at the completion of a shed command execution, are as specified in the individual property descriptions.

The Load Control object shed mechanism follows a state machine whose operation is displayed in Figure 12-2. This state machine only describes the behavior of the Load Control object when the Enable property has the value TRUE. See Clause 12.19.14 for a description of the effect of this property. The state machine captures the transitions that occur within the Load Control object.

If the device is unable to comply fully with the shed request by shedding the entire amount of load requested, it is a local matter whether the device sheds as much load as it can or whether it does not shed any of its loads. Determination of compliance with a client's load shed request may also be affected by other factors, such as the definition of the baseline usage, synchronization of time between the client and the device containing the Load Control object, and any intrinsic limits on shed amounts that the device may have. If these factors are not in agreement, the client's determination of compliance may not match the object's determination.

The activity of a Load Control object in the SHED_REQUEST_PENDING state will vary. For a Load Control object controlling only one or more direct loads that it can shed instantly, the activity will be simply waiting for the first duty window to arrive, at which point it will monitor the clock and cycle on/off or begin modulation of loads or reduce loads by some other means. The object may need to begin shedding some of the loads before Start_Time in order to meet the shed target by Start_Time, in which case it will enter the SHED_NON_COMPLIANT state. For a Load Control object controlling other Load Control objects subordinate to it, the shedding activity will begin prior to Start_Time by communicating the shed request (possibly modified) to these other Load Control objects. There may be some Load Control objects that indicate an inability to comply with the request, which may lead to requests for increased load reduction from these or other Load Control objects.

While the Load Control object is designed to allow independent operation, it is possible that there will exist within a building (or even within a device) a hierarchy of Load Control objects, where one Load Control object receives a load shed command, possibly from a non-BACnet client (e.g., a utility), and the controller which hosts that object (the master) in turn will be responsible for managing and issuing requests to other Load Control objects. There may be a negotiation between the master and its subordinate Load Control objects. The master uses WriteProperty or WritePropertyMultiple to set shed request parameters in the subordinate Load Control objects. A subordinate Load Control object would then set its Expected_Shed_Level property to the value that it expects to be able to achieve after Start_Time. Before

Start_Time, the master object can read the Expected_Shed_Level properties of its subordinates to determine expected compliance with the request. After Start_Time plus Duty_Window, the Actual_Shed_Level properties of the subordinate objects will reflect the actual amount shed in the past Duty_Window. If by reading these properties the master Load Control object determines that one or more subordinate objects cannot completely comply with the request, the master may choose to modify the shed requests to subordinates, such that the overall shed target is achieved. For instance, it may request that another object shed a greater amount of its load or it may choose to request that the noncompliant device shed a greater amount. This negotiation could be repeated at each successive level in the hierarchy. If the subordinate Load Control objects also support intrinsic reporting, expected or actual instances of non-compliance can be reported to the master object using event notifications.

Where large loads are concerned, it is expected that the master Load Control object will employ sequencing to distribute the startup and shutdown of managed loads. When the load control master is used in a gateway to a non-BACnet load control client, such as a utility company, the gateway shall accept and process any start randomization commands and accordingly distribute the initiation of load control requests to its subordinate Load Control objects.

The Load Control object shall exhibit restorative behavior across a restart or time change of the BACnet device in which it resides. The shed request property values shall be maintained across a device restart. Upon device restart or a time change, the object shall behave as if Start_Time were written and shall re-evaluate the state machine's state.

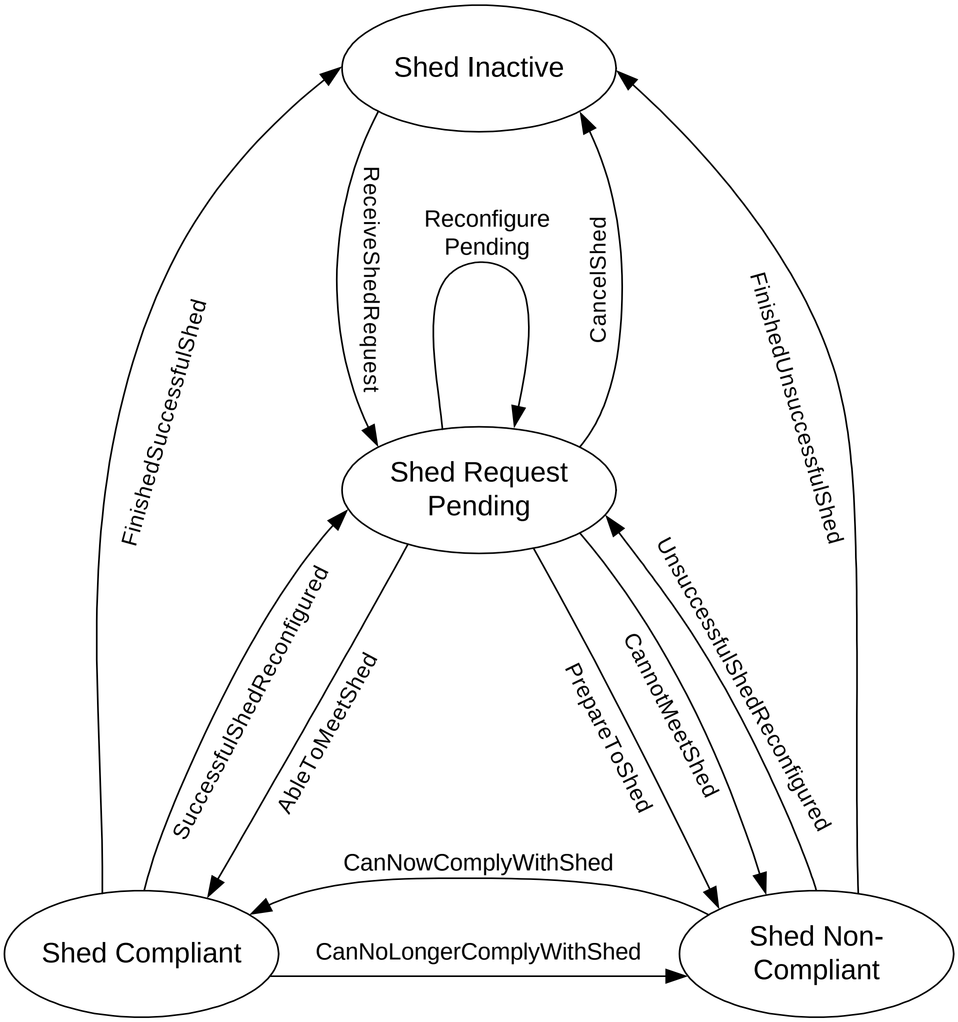

Figure 12-2. State Diagram for Load Control Object.

SHED_INACTIVE

In the SHED_INACTIVE state, the Load Control object waits for a shed request.

ReceiveShedRequest

If Start_Time is written, the object shall calculate Expected_Shed_Level and Actual_Shed_Level and enter the SHED_REQUEST_PENDING state.

SHED_REQUEST_PENDING

In the SHED_REQUEST_PENDING state, the object makes a determination from the newly written shed parameters whether the shed request needs to be executed immediately or at some time in the future.

CancelShed

If the current time is after Start_Time plus Shed_Duration, this request is for an invalid time and is ignored. The object shall stop shedding and enter the SHED_INACTIVE state.

If Requested_Shed_Level is equal to the default value for the choice, or Start_Time contains wildcards, then this is a cancellation of shedding. The object shall stop shedding and enter the SHED_INACTIVE state.

ReconfigurePending

If the current time is prior to Start_Time, and a new write is received for Requested_Shed_Level, Shed_Duration, Duty_Window, or Start_Time, this is a reconfiguration of the shed request. The object shall calculate Expected_Shed_Level and Actual_Shed_Level and enter the SHED_REQUEST_PENDING state.

PrepareToShed

If the current time is prior to Start_Time, but the loads to be shed require time to decrease usage to the requested shed level, the object may choose to initiate shedding of its subordinates prior to Start_Time in order to be in compliance by Start_Time. If this approach is followed, the object shall calculate Expected_Shed_Level and Actual_Shed_Level and enter the SHED_NON_COMPLIANT state.

CannotMeetShed

If the current time is after Start_Time, and the object is unable to meet the shed request immediately, it shall begin shedding its loads, calculate Expected_Shed_Level and Actual_Shed_Level, and enter the SHED_NON_COMPLIANT state.

AbleToMeetShed

If the current time is after Start_Time and the object is able to achieve the shed request immediately, it shall shed its loads, calculate Expected_Shed_Level and Actual_Shed_Level, and enter the SHED_COMPLIANT state.

If the current time is before Start_Time, and the object has initiated shedding prior to Start_Time in order to be in compliance by Start_Time, and the object has achieved the requested shed level, it shall calculate Expected_Shed_Level and Actual_Shed_Level and enter the SHED_COMPLIANT state.

SHED_NON_COMPLIANT

In the SHED_NON_COMPLIANT state, the object attempts to meet the shed request until the shed is achieved, the object is reconfigured, or the request has completed unsuccessfully.

FinishedUnsuccessfulShed

If the current time is after Start_Time plus Shed_Duration, the shed request has completed unsuccessfully. The object shall stop shedding and enter the SHED_INACTIVE state.

UnsuccessfulShedReconfigured

If the object receives a write to any of the properties Requested_Shed_Level, Shed_Duration, Duty_Window, or Start_Time, the object shall enter the SHED_REQUEST_PENDING state.

CanNowComplyWithShed

If the object has achieved the Requested_Shed_Level, it shall calculate Expected_Shed_Level and Actual_Shed_Level and enter the SHED_COMPLIANT state.

SHED_COMPLIANT

In the SHED_COMPLIANT state, the object continues meeting the shed request until the shed is either reconfigured or completes, or conditions change and the object is no longer able to maintain the requested shed level.

FinishedSuccessfulShed

If the current time is after Start_Time plus Shed_Duration, the shed request has completed successfully. The object shall stop shedding, set Start_Time to all wildcards, and enter the SHED_INACTIVE state.

SuccessfulShedReconfigured

If the object receives a write to any of the properties Requested_Shed_Level, Shed_Duration, Duty_Window, or Start_Time, the object shall enter the SHED_REQUEST_PENDING state.

CanNoLongerComplyWithShed

If the object is no longer able to maintain the Requested_Shed_Level, it shall calculate Expected_Shed_Level and Actual_Shed_Level and enter the SHED_NON_COMPLIANT state.

Table 12-20. Properties of the Load Control Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	BACnetShedState	R
State_Description	CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	R
Reliability	BACnetReliability	O
Requested_Shed_Level	BACnetShedLevel	W
Start_Time	BACnetDateTime	W
Shed_Duration	Unsigned	W
Duty_Window	Unsigned	W
Enable	BOOLEAN	W
Full_Duty_Baseline	REAL	O
Expected_Shed_Level	BACnetShedLevel	R
Actual_Shed_Level	BACnetShedLevel	R
Shed_Levels	BACnetARRAY[N] of Unsigned	W[1]
Shed_Level_Descriptions	BACnetARRAY[N] of CharacterString	R
Notification_Class	Unsigned	O[2]
Time_Delay	Unsigned	O[2]
Event_Enable	BACnetEventTransitionBits	O[2]
Acked_Transitions	BACnetEventTransitionBits	O[2]
Notify_Type	BACnetNotifyType	O[2]
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	O[2]
Profile_Name	CharacterString	O

[1] The elements of this array are required to be writable, although the array is not required to be resizable.

[2] These properties are required if the object supports intrinsic reporting.

12.17.1 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.

12.17.2 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.

12.17.3 Object_Type

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

12.17.4 Description

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

12.17.5 Present_Value

This property, of type BACnetShedState, indicates the current load shedding state of the object. See Figure 12-2 for a diagram of the state machine governing the value of Present_Value.

12.17.6 State_Description

This property, of type CharacterString, is a string of printable characters whose content is not restricted. The State_Description provides additional information for human operators about the shed state of the Load Control object.

12.17.7 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a Load Control object. 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 point has been overridden by some mechanism local to the BACnet Device, otherwise logical FALSE (0).

OUT_OF_SERVICE Logical TRUE (1) if the Out_Of_Service property has a value of TRUE, otherwise logical FALSE (0).

12.17.8 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. If the object supports intrinsic reporting, then the Event_State property shall indicate the event state of the object. If the object does not support intrinsic reporting and if the Reliability property is not present, then the value of this property shall be NORMAL. If the Reliability property is present and does not have a value of NO_FAULT_DETECTED, then the value of the Event_State property shall be FAULT. Changes in the Event_State property to the value FAULT are considered to be “fault” events.

12.17.9 Reliability

The Reliability property, of type BACnetReliability, provides an indication of whether the Load Control object is reliably reporting its compliance with any load shed requests. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, UNRELIABLE_OTHER}

12.17.10 Requested_Shed_Level

This property, of type BACnetShedLevel, indicates the desired load shedding. Table 12-21 describes the default values and power targets for the different choices of Requested_Shed_Level.

If the choice for Requested_Shed_Level is PERCENT, the value of Requested_Shed_Level is interpreted as a requested percentage of Full Duty to which the device is to attempt to reduce its load. The determination of the Full Duty rating (or some alternative baseline power usage) is a local matter. It may be determined from the Full_Duty_Baseline property, if present.

If the choice for Requested_Shed_Level is LEVEL, the value of Requested_Shed_Level is used to set a preconfigured level of load shedding.

The Load Control object’s available shed actions are described by the Shed_Level_Descriptions array and are mapped to the BACnet visible values of Requested_Shed_Level by the Shed_Levels array. The SHED_INACTIVE state shall always be represented by the value 0, which is not represented in the Shed_Levels or Shed_Level_Descriptions arrays. If Requested_Shed_Level choice is AMOUNT, the value of Requested_Shed_Level shall be interpreted as an amount, in kilowatts, by which to reduce power usage. Load Control objects are required to support the LEVEL choice. Support for the PERCENT and AMOUNT choices is optional. This allows a master to be guaranteed the ability to write to the Load Control object by using the LEVEL choice.

If a load control command has been issued, and execution of the command has completed, Requested_Shed_Level shall be reset to the default value appropriate to the choice of Requested_Shed_Level used for the last command.

Table 12-21. Requested_Shed_Level Default Values and Power Targets

Choice	Default Requested_Shed_Level value	Power load target in kW
PERCENT	100	(current baseline) * Requested_Shed_Level / 100
LEVEL	0	locally pre-specified shed target for the given level
AMOUNT	0.0	(current baseline) - Requested_Shed_Level

12.17.11 Start_Time

This property, of type BACnetDateTime, indicates the start of the duty window in which the load controlled by the Load Control object must be compliant with the requested shed. Load shedding (or determination of loads to shed) may need to begin before Start_Time in order to be compliant with the shed request by Start_Time. If no shed request is pending or active, Start_Time shall contain all wildcard values. If a load control command has been issued, and execution of the command has completed, Start_Time shall be reset by the device to contain all wildcard values.

12.17.12 Shed_Duration

This property, of type Unsigned, indicates the duration of the load shed action, starting at Start_Time. The units for Shed_Duration are minutes. If no shed request is pending or active, Shed_Duration shall contain wildcard values. If a load control command has been issued, and execution of the command has completed, Shed_Duration shall be reset by the device to contain wildcard values. A load control command may be issued without Shed_Duration being set to indicate an indefinite length load shed request.

12.17.13 Duty_Window

This property, of type Unsigned, indicates the time window used for load shed accounting. The units for Duty_Window are minutes. Duty_Window is used for performance measurement or compliance purposes. The average power consumption across a duty window must be less than or equal to the requested reduced consumption. It is a local matter whether this window is fixed or sliding. The first Duty_Window begins at Start_Time. If a shed request is received with no value written to this property, Duty_Window shall be set to some pre-agreed upon value. If a load control command has been issued, and execution of the command has completed, Duty_Window shall be reset by the device to this pre-agreed value.

12.17.14 Enable

This property, of type BOOLEAN, indicates and controls whether the Load Control object is currently enabled to respond to load shed requests. If Enable is TRUE, the object will respond to load shed requests normally and follow the state machine described in Figure 12-2. If Enable is FALSE, the object will transition to the SHED_INACTIVE state if necessary and remain in that state. It shall not respond to any load shed request while Enable is FALSE.

12.17.15 Full_Duty_Baseline

This property, of type REAL, indicates the baseline power consumption value for this device, if a fixed baseline is used. Shed requests may be made with respect to this baseline, that is , to “percent of baseline” and “amount off baseline”. The units of Full_Duty_Baseline are kilowatts.

12.17.16 Expected_Shed_Level

This property, of type BACnetShedLevel, indicates the amount of power that the object expects to be able to shed in response to a load shed request. When the object is in the SHED_INACTIVE state, this value shall be equal to the default value of Requested_Shed_Level. When a shed request is pending or active, Expected_Shed_Level shall be equal to the shed level the object expects to be able to achieve at Start_Time. Expected_Shed_Level allows a client (e.g., a master-level Load Control object) to determine if a pending shed request needs to be modified in order to achieve the requested shed level, in the event that Expected_Shed_Level is less than the Requested_Shed_Level. The units for Expected_Shed_Level are the same as the units for Requested_Shed_Level.

12.17.17 Actual_Shed_Level

This property, of type BACnetShedLevel, indicates the actual amount of power being shed in response to a load shed request. When the object is in the SHED_INACTIVE state, this value shall be equal to the default value of Requested_Shed_Level. After Start_Time plus Duty_Window has elapsed, this value shall be the actual shed amount as calculated based on the average value over the previous duty window. The units for Actual_Shed_Level are the same as the units for Requested_Shed_Level.

12.17.18 Shed_Levels

This property is a BACnetARRAY of unsigned integers representing the shed levels for the LEVEL choice of BACnetShedLevel that have meaning for this particular Load Control object. When commanded with the LEVEL choice, the Load Control object shall take a shedding action described by the corresponding element in the Shed_Level_Descriptions array. If the Load Control object is commanded to go to a level that is not in the Shed_Levels array, it shall go to the level with the nearest lower value defined in Shed_Levels. The elements of the array are required to be writable, allowing local configuration of how this Load Control object will participate in load shedding for the facility. This array is not required to be resizable through BACnet write services. The size of this array shall be equal to the size of the Shed_Level_Descriptions array.

12.17.19 Shed_Level_Descriptions

This property is a BACnetARRAY of character strings representing a description of the shed levels that the Load Control object can take on. This allows a local configuration tool to provide to a user an understanding of what each shed level in this Load Control object's load shedding algorithm will do. The level at which each shed action will occur can then be configured by writing to the Shed_Levels property.

12.17.20 Notification_Class

This 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.

12.17.21 Time_Delay

This property, of type Unsigned, shall specify the minimum period of time in seconds that the Present_Value property shall remain equal to SHED_NON_COMPLIANT when the current time is after Start_Time and before a TO-OFFNORMAL event is generated, or not equal to SHED_NON_COMPLIANT before a TO-NORMAL event is generated. This property is required if intrinsic reporting is supported by this object.

12.17.22 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.

12.17.22.1 Conditions for Generating a TO-OFFNORMAL Event

A TO-OFFNORMAL event is generated under these conditions:

(a) the Present_Value must remain in the SHED_NON_COMPLIANT state for a minimum period of time, specified in the Time_Delay property,
(b) the TO-OFFNORMAL flag must be set in the Event_Enable property, and
(c) the current time is after Start_Time.

12.17.22.2 Conditions for Generating a TO-NORMAL Event

Once the Present_Value transitions to the SHED_NON_COMPLIANT state, a TO-NORMAL event is generated under these conditions:

(a) the Present_Value leaves the SHED_NON_COMPLIANT state and remains out of this state for a minimum period of time, specified in the Time_Delay property, and
(b) the TO-NORMAL flag must be set in the Event_Enable property.

12.17.23 Acked_Transitions

This property, of type BACnetEventTransitionBits, shall convey three flags that separately indicate the receipt of acknowledgements 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 acknowledgement;

(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 acknowledgement is expected); or

(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 acknowledgement is expected).

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

12.17.24 Notify_Type

This 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.

12.17.25 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.

12.17.26 Profile_Name

This 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.

[Change Table **13-1**, p.254]

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

Object Type	Criteria	Properties Reported
...		
Loop	If Present_Value changes by COV_Increment or Status_Flags changes at all	Present_Value, Status_Flags, Setpoint, Controlled_Variable_Value
Load Control	*If Present_Value, Requested_Shed_Level, Start_Time, Shed_Duration, or Duty_Window changes at all*	*Present_Value, Status_Flags, Requested_Shed_Level, Start_Time, Shed_Duration, Duty_Window*
Pulse Converter	If Present_Value changes by COV_Increment or Status_Flags changes at all or If COV_Period expires	Present_Value, Status_Flags, Update_Time
…		

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

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

Object Type	Criteria	Event Type
...		
Binary Output, Multi-state Output	If Present_Value differs from Feedback_Value for longer than Time_Delay AND the new transition is enabled in Event_Enable	COMMAND_FAILURE
Load Control	*If Present_Value equals SHED_NON_COMPLIANT for longer than Time_Delay AND the new transition is enabled in Event_Enable*	*COMMAND_FAILURE*
Loop	If the absolute difference between Setpoint and Controlled_Variable_Value exceeds Error_Limit for longer than Time_Delay AND the new transition is enabled in Event_Enable	FLOATING_LIMIT
...		

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

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

Object	Event Type	Notification Parameters	Referenced Object's Properties
...			
Binary Output, Multi-state Output	COMMAND_FAILURE	Command_Value Status_Flags Feedback_Value	Present_Value Status_Flags Feedback_Value
Load Control	*COMMAND_FAILURE*	*Command_Value* *Status_Flags* *Feedback_Value*	*Requested_Shed_Level* *Status_Flags* *Actual_Shed_Level*
Loop	FLOATING_LIMIT	Referenced_Value Status_Flags Setpoint_Value Error_Limit	Controlled_Variable_Value Status_Flags Setpoint Error_Limit
...			

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

```
BACnetObjectType ::= ENUMERATED {
    ...
    life-safety-zone            (22),
    load-control                (28),
    ...
    -- enumeration 27 is used in Addendum b to ANSI/ASHRAE 135-2004(135b-3)
    -- see load-control         (28),
    -- enumeration 29 is used in Addendum d to ANSI/ASHRAE 135-2004(135d-1)
    ...
    }
```

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

```
BACnetObjectTypesSupported ::= ENUMERATED {
    ...
    -- life-safety-zone          (22),
    -- load-control              (28),
    ...
    -- Objects added after 2004
    -- enumerations 25-27 are used in Addendum b to ANSI/ASHRAE 135-2004(135b-1, -2 and -3)
     load-control                (28),
    -- enumeration 29 is used in Addendum d to ANSI/ASHRAE 135-2004(135d-1)
    }
```

[Add to **21**, **BACnetPropertyIdentifier** production (distributed alphabetically), pp. 423-428]

```
BACnetPropertyIdentifier ::= ENUMERATED {
    ...
    active-cov-subscriptions                (152),
    actual-shed-level                       (212),
    ...
    direct-reading                          (156),
    duty-window                             (213),
    ...
    -- The enable property appears in Addendum b to ANSI/ASHRAE 135-2004(135b-2)
    ...
    exception-schedule                      (38),
    expected-shed-level                     (214),
    ...
    firmware-revision                       (44),
    full-duty-baseline                      (215),
    ...
    relinquish-default                      (104),
    requested-shed-level                    (218),
    ...
    setting                                 (162),
    shed-duration                           (219),
    shed-level-descriptions                 (220),
    shed-levels                             (221),
    …
    start-time                              (142),
    …
    -- enumerations 193-206 are used in Addendum b to ANSI/ASHRAE 135-2004(135b-2)
    -- enumerations 207-211are  used in Addendum d to ANSI/ASHRAE 135-2004(135d-1)
    -- see actual-shed-level                (212),
    -- see duty-window                      (213),
    -- see expected-shed-level              (214),
    -- see full-duty-baseline               (215),
    -- see node-subtype                     (216),
    -- see node-type                        (217),
    -- see requested-shed-level             (218),
    -- see shed-duration                    (219),
    -- see shed-level-descriptions          (220),
    -- see shed-levels                      (221),
    -- see state-description                (222),
    -- see structured-object-list           (223),
    -- see subordinate-annotations          (224),
    -- see subordinate-list                    (225),
    ...
    }
```

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

[Add to **21**, new **BACnetShedLevel** production, p. 431]

```
BACnetShedLevel :: = CHOICE {
    percent        [0]  Unsigned,
    level          [1]  Unsigned,
    amount         [2]  REAL
    }
```

[Add to **21**, new **BACnetShedState** enumeration, p. 431]

```
BACnetShedState ::= ENUMERATED {
    shed-inactive               (0),
    shed-request-pending        (1),
    shed-compliant              (2),
    shed-non-compliant          (3)
    }
```

[Add to **Annex C**, p. 460]

```
LOAD-CONTROL ::= SEQUENCE {
    object-identifier           [75]   BACnetObjectIdentifier,
    object-name                 [77]   CharacterString,
    object-type                 [79]   BACnetObjectType,
    description                 [28]   CharacterString OPTIONAL,
    present-value               [85]   BACnetShedState,
    state-description           [28]   CharacterString OPTIONAL,
    status-flags                [111]  BACnetStatusFlags,
    event-state                 [36]   BACnetEventState,
    reliability                 [103]  BACnetReliability OPTIONAL,
    requested-shed-level        [218]  BACnetShedLevel,
    start-time                  [142]  BACnetDateTime,
    shed-duration               [219]  Unsigned,
    duty-window                 [213]  Unsigned,
    enable                      [133]  BOOLEAN,
                                -- The enable  property appears in Addendum b to ANSI/ASHRAE 135-2004(135b-2)
    full-duty-baseline          [215]  REAL OPTIONAL,
    expected-shed-level         [214]  BACnetShedLevel,
    actual-shed-level           [212]  BACnetShedLevel,
    shed-levels                 [221]  SEQUENCE OF Unsigned, -- accessed as a BACnetARRAY
    shed-level-descriptions     [220]  SEQUENCE OF CharacterString, -- accessed as a BACnetARRAY
    notification-class          [17]   Unsigned OPTIONAL,
    time-delay                  [113]  Unsigned 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             [167] CharacterString OPTIONAL
        }
```

[Add new **Annex D.17**, p. 477, and renumber existing **Annex D.17** and subsequent examples.]

D.17 Example of a Load Control object

Property:	Object_Identifier =	(Load_Control, Instance 1)
Property:	Object_Name =	"Load Control 1"
Property:	Object_Type =	LOAD CONTROL
Property:	Description =	"Chiller Load Control"
Property:	Present_Value =	SHED_COMPLIANT
Property:	State_Description =	"shedding compliant with request"
Property:	Status_Flags =	{FALSE, FALSE, FALSE, FALSE}
Property:	Event_State =	NORMAL
Property:	Reliability =	NO_FAULT_DETECTED
Property:	Requested_Shed_Level =	(Percent, 80)
Property:	Start_Time =	(1-JUL-2002,10:00:00.0)
Property:	Shed_Duration =	120
Property:	Duty_Window =	30
Property:	Enabled =	TRUE
Property:	Full_Duty_Baseline =	250.0
Property:	Expected_Shed_Level =	80
Property:	Actual_Shed_Level =	80
Property:	Shed_Levels =	(1, 3, 6, 9)
Property:	Shed_Level_Descriptions =	(“switch to alternate energy”, “setback 2 degrees”, “dim lights 20%”, “setback 4 degrees”)
Property:	Notification_Class =	3
Property:	Time_Delay =	60
Property:	Event_Enable =	{TRUE, FALSE, TRUE}
Property:	Acked_Transitions =	{TRUE, TRUE, TRUE}
Property:	Notify_Type =	ALARM
Property:	Event_Time_Stamps =	((17-APR-2001,18:50:21.2), (*-*-*,*:*:*.*), (17-APR-2001,19:01:34.0))