ANSI/ASHRAE Addendum w
to ANSI/ASHRAE Standard 135-2008

ASHRAE STANDARD

BACnet®—A Data Communication Protocol for Building Automation and Control Networks

Approved by the ASHRAE Standards Committee on January 23, 2010; by the ASHRAE Board of Directors on January 27, 2010; and by the American National Standards Institute on January 28, 2010.

This standard is under continuous maintenance by a Standing Standard Project Committee (SSPC) for which the Standards Committee has established a documented program for regular publication of addenda or revisions, including procedures for timely, documented, consensus action on requests for change to any part of the standard. The change submittal form, instructions, and deadlines may be obtained in electronic form from the ASHRAE Web site, http://www.ashrae.org, or in paper form from the Manager of Standards. The latest edition of an ASHRAE Standard may be purchased from ASHRAE Customer Service, 1791 Tullie Circle, NE, Atlanta, GA 30329-2305. E-mail: orders@ashrae.org. Fax: 404-321-5478. Telephone: 404-636-8400 (worldwide), or toll free 1-800-527-4723 (for orders in US and Canada).

ISSN 1041-2336

Approved American National Standard
ANSI
www.ansi.org

American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.
1791 Tullie Circle NE, Atlanta, GA 30329
www.ashrae.org

SPECIAL NOTE

This American National Standard (ANS) is a national voluntary consensus standard developed under the auspices of the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). *Consensus* is defined by the American National Standards Institute (ANSI), of which ASHRAE is a member and which has approved this standard as an ANS, as "substantial agreement reached by directly and materially affected interest categories. This signifies the concurrence of more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that an effort be made toward their resolution." Compliance with this standard is voluntary until and unless a legal jurisdiction makes compliance mandatory through legislation.

ASHRAE obtains consensus through participation of its national and international members, associated societies, and public review.

ASHRAE Standards are prepared by a Project Committee appointed specifically for the purpose of writing the Standard. The Project Committee Chair and Vice-Chair must be members of ASHRAE; while other committee members may or may not be ASHRAE members, all must be technically qualified in the subject area of the Standard. Every effort is made to balance the concerned interests on all Project Committees.

The Manager of Standards of ASHRAE should be contacted for:

a. interpretation of the contents of this Standard,
b. participation in the next review of the Standard,
c. offering constructive criticism for improving the Standard, or
d. permission to reprint portions of the Standard.

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

FOREWORD

Addendum 135*w* to ANSI/ASHRAE Standard 135-2008 contains a number of changes to the current standard. These modifications are the result of change proposals made pursuant to the ASHRAE continuous maintenance procedures and of deliberations within Standing Standard Project Committee 135. The changes are summarized below.

135-2008*w*-1. Add more primitive value objects, p. 2.

In the following document, language added to existing clauses of ANSI/ASHRAE 135-2008 and addenda is indicated through the use of *italics*, while deletions are indicated by ~~strikethrough~~. Where entirely new subclauses are added, plain type is used throughout.

135-2008*w*-1. Add more primitive value objects.

Rationale
While the existing standard objects provide many useful types of basic objects, in practice there are some kinds of data values that are not conveniently represented as a BACnet standard object type. This proposed change defines objects for all the primitive data types.

Addendum 135-2008*w*-1

[Add new Clauses **12.A through 12.L**, p. 288]

12.A CharacterString Value Object Type

The CharacterString Value object type defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a CharacterString Value object to make any kind of character string data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

If a set of strings is known and fixed, then a MultiState Value object is an alternative that may provide some benefit to automated processes consuming the numeric Present_Value.

Table 12-A. Properties of the CharacterString Value Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	CharacterString	R[1]
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	O[3]
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	O
Priority_Array	BACnetPriorityArray	O[2]
Relinquish_Default	CharacterString	O[2]
Time_Delay	Unsigned	O[3]
Notification_Class	Unsigned	O[3]
Alarm_Values	BACnetARRAY[N] of BACnetOptionalCharacterString	O[3]
Fault_Values	BACnetARRAY[N] of BACnetOptionalCharacterString	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] If Present_Value is commandable, then it is required to be writable. This property is required to be writable when Out_Of_Service is TRUE.
[2] If Present_Value is commandable, then both of these properties shall be present.
[3] These properties are required if the object supports intrinsic reporting.

12.A.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.A.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.A.3 Object_Type

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

12.A.4 Description

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

12.A.5 Present_Value

This property, of type CharacterString, indicates the current value of the object. The Present_Value property shall be writable when Out_Of_Service is TRUE (see Clause 12.A.9).

12.A.6 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a CharacterString Value object. Three of the flags are associated with the values of another property of this object. A more detailed status could be determined by reading the property that is linked to this flag. 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 TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).
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. In this context "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. Otherwise, the value is logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property is present and has a value of TRUE, otherwise logical FALSE (0).

12.A.7 Event_State

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

12.A.8 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the CharacterString Value object is reliably reporting its value. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, COMMUNICATION_FAILURE, MULTI_STATE_FAULT, UNRELIABLE_OTHER}

12.A.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.A.9 Out_Of_Service

This optional property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present_Value of the CharacterString Value object is decoupled from software local to the BACnet device in which the object resides that normally produces the Present_Value as an output or consumes it as an input. When Out_Of_Service is TRUE, the Present_Value property may be written to freely.

12.A.10 Priority_Array

This optional property, of type BACnetPriorityArray, is a read-only array containing prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism. If either the Priority_Array property or the Relinquish_Default property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.A.11 Relinquish_Default

This optional 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. If either the Relinquish_Default property or the Priority_Array property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.A.12 Time_Delay

This optional property, of type Unsigned, shall specify the minimum period of time in seconds during which the Present_Value must contain one of the character strings specified by the Alarm_Values property before a TO-OFFNORMAL event is generated, or not contain any of the non-empty character strings specified by the Alarm_Values property before a TO-NORMAL event is generated. This property is required if intrinsic reporting is supported by this object.

12.A.13 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.A.14 Alarm_Values

This optional property, of type BACnetARRAY of BACnetOptionalCharacterString, shall specify one or more string values, at least one of which must match Present_Value before an event is generated. A "match" with Present_Value is defined as follows:

a. If the alarm string is NULL, it is not considered a match.
b. If the alarm string is empty (of zero length), then it is considered a match if and only if the Present_Value is also an empty string.
c. If the alarm string is not empty, then it is considered a match if the alarm string appears in any position within the Present_Value string. For character-matching purposes, character case shall be significant, and so a match must be an exact match character by character.

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

12.A.14.1 Conditions for Generating a TO-OFFNORMAL Event

A TO-OFFNORMAL event is generated under these conditions:

(a) the Present_Value must match at least one of the string values in the Alarm_Values list, for a minimum period of time, specified in the Time_Delay property, and
(b) the TO-OFFNORMAL flag must be enabled in the Event_Enable property.

After a TO-OFFNORMAL event has been generated and before a TO-NORMAL event has been generated, if the Present_Value changes such that it matches a different string from the Alarm_Values list for Time_Delay seconds, the algorithm shall generate another TO-OFFNORMAL event.

12.A.14.2 Conditions for Generating a TO-NORMAL Event

Once matching a value in this property, the Present_Value must not match any of the values in this property before a TO-NORMAL event is generated under these conditions:

(a) the Present_Value must remain not matching any of the non-empty string values in Alarm_Values for a minimum period of time, specified by the Time_Delay property, and
(b) the TO-NORMAL flag must be enabled in the Event_Enable property.

12.A.15 Fault_Values

This optional property, of type BACnetARRAY of BACnetOptionalString, shall specify any value the Present_Value must match before a TO-FAULT event is generated. If Present_Value matches any of the states in the Fault_Values list, then the Reliability property shall have the value MULTI_STATE_FAULT. The Fault_Values property is required if intrinsic reporting is supported by this object.

The method for "matching" fault vaues is the same as for Alarm_Values. See 12.A.14

12.A.16 Event_Enable

This optional 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.A.17 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 that 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 that no acknowledgment is expected).

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

12.A.18 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.A.19 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.A.20 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.

12.B DateTime Value Object Type

The DateTime Value object type defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a DateTime Value object to make any kind of datetime data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

A DateTime Value object is used to represent a single moment in time. In contrast, the DateTime Pattern Value object can be used to represent multiple recurring dates and times.

Table 12-B. Properties of the DateTime Value Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	BACnetDateTime	R[1]
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	O
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	O
Priority_Array	BACnetPriorityArray	O[2]
Relinquish_Default	BACnetDateTime	O[2]
Is_UTC	BOOLEAN	O
Profile_Name	CharacterString	O

[1] If Present_Value is commandable, then it is required to be writable. This property is required to be writable when Out_Of_Service is TRUE.
[2] If Present_Value is commandable, then both of these properties shall be present.

12.B.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.B.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.B.3 Object_Type

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

12.B.4 Description

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

12.B.5 Present_Value

This property, of type BACnetDateTime, indicates the current value of the object. The value of this property shall contain either a fully specified date and time or it shall indicate a fully unspecified date and time by setting all octets to X'FF'. A fully specified date and time does not contain any octets that are equal to X'FF' or the special values for the 'month' or 'day of month' fields, the Present_Value property shall be writable when Out_Of_Service is TRUE (see Clause 12.B.9).

12.B.6 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a DateTime Value object. Three of the flags are associated with the values of another property of this object. A more detailed status could be determined by reading the property that is linked to this flag. 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 TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).
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. In this context "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. Otherwise, the value is logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property is present and has a value of TRUE, otherwise logical FALSE (0).

12.B.7 Event_State

This optional 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 Reliability property is not present, or if the Reliability property is present and has a value of NO_FAULT_DETECTED, 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.

12.B.8 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the DateTime Value object is reliably reporting its value. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, COMMUNICATION_FAILURE, UNRELIABLE_OTHER}

12.B.9 Out_Of_Service

This optional property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present_Value of the DateTime Value object is decoupled from software local to the BACnet device in which the object resides that normally produces the Present_Value as an output or consumes it as an input. When Out_Of_Service is TRUE, the Present_Value property may be written to freely.

12.B.10 Priority_Array

This optional property is a read-only array containing prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism. If either the Priority_Array property or the Relinquish_Default property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.B.11 Relinquish_Default

This optional property, of type BACnetDateTime, 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. If either the Relinquish_Default property or the Priority_Array property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.B.12 Is_UTC

This optional property, of type BOOLEAN, indicates whether the Present_Value property indicates a UTC date and time (when TRUE) or a local date and time (when FALSE). If this property is absent, the Present_Value shall be a local date and time.

12.B.13 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.

12.C Large Analog Value Object Type

The Large Analog Value object type defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a Large Analog Value object to make any kind of double-precision data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

Table 12-C. Properties of the Large Analog Value Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	Double	R[1]
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	O[4]
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	O
Units	BACnetEngineeringUnits	R
Priority_Array	BACnetPriorityArray	O[2]
Relinquish_Default	Double	O[2]
COV_Increment	Double	O[3]
Time_Delay	Unsigned	O[4]
Notification_Class	Unsigned	O[4]
High_Limit	Double	O[4]
Low_Limit	Double	O[4]
Deadband	Double	O[4]
Limit_Enable	BACnetLimitEnable	O[4]
Event_Enable	BACnetEventTransitionBits	O[4]
Acked_Transitions	BACnetEventTransitionBits	O[4]
Notify_Type	BACnetNotifyType	O[4]
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	O[4]
Profile_Name	CharacterString	O

[1] If Present_Value is commandable, then it is required to be writable. This property is required to be writable when Out_Of_Service is TRUE.
[2] If Present_Value is commandable, then both of these properties shall be present.
[3] This property is required if the object supports COV reporting.
[4] These properties are required if the object supports intrinsic reporting.

12.C.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.C.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.C.3 Object_Type

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

12.C.4 Description

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

12.C.5 Present_Value

This property, of type Double, indicates the current value of the object. The Present_Value property shall be writable when Out_Of_Service is TRUE (see Clause 12.C.9).

12.C.6 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a Large Analog Value object. Three of the flags are associated with the values of another property of this object. A more detailed status could be determined by reading the property that is linked to this flag. 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 TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).
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. In this context "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. Otherwise, the value is logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property is present and has a value of TRUE, otherwise logical FALSE (0).

12.C.7 Event_State

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

12.C.8 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the Large Analog Value object is reliably reporting its value. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, COMMUNICATION_FAILURE, UNRELIABLE_OTHER}

12.C.9 Out_Of_Service

This optional property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present_Value of the Large Analog Value object is decoupled from software local to the BACnet device in which the object resides that normally produces the Present_Value as an output or consumes it as an input. When Out_Of_Service is TRUE, the Present_Value property may be written to freely.

12.C.10 Units

This property, of type BACnetEngineeringUnits, indicates the measurement units of this object. See the BACnetEngineeringUnits ASN.1 production in Clause 21 for a list of engineering units defined by this standard.

12.C.11 Priority_Array

This optional property, of type BACnetPriorityArray, is a read-only array containing prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism. If either the Priority_Array property or the Relinquish_Default property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.C.12 Relinquish_Default

This optional property, of type Double, 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. If either the Relinquish_Default property or the Priority_Array property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.C.13 COV_Increment

This optional property, of type Double, shall specify the minimum change in Present_Value that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

12.C.14 Time_Delay

This optional property, of type Unsigned, shall specify the minimum period of time in seconds during which the Present_Value must remain outside the band defined by the High_Limit and Low_Limit properties before a TO-OFFNORMAL event is generated, or within the same band, including the Deadband property, before a TO-NORMAL event is generated. This property is required if intrinsic reporting is supported by this object.

12.C.15 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.C.16 High_Limit

This optional property, of type Double, shall specify a limit that the Present_Value must exceed before an event is generated. This property is required if intrinsic reporting is supported by this object.

12.C.16.1 Conditions for Generating a TO-OFFNORMAL Event

A TO-OFFNORMAL event is generated under these conditions:

(a) the Present_Value must exceed the High_Limit for a minimum period of time, specified in the Time_Delay property, and
(b) the HighLimitEnable flag must be set in the Limit_Enable property, and
(c) the TO-OFFNORMAL flag must be set in the Event_Enable property.

12.C.16.2 Conditions for Generating a TO-NORMAL Event

Once exceeded, the Present_Value must fall below the High_Limit minus the Deadband before a TO-NORMAL event is generated under these conditions:

(a) the Present_Value must fall below the High_Limit minus the Deadband for a minimum period of time, specified in the Time_Delay property, and
(b) the HighLimitEnable flag must be set in the Limit_Enable property, and
(c) the TO-NORMAL flag must be set in the Event_Enable property.

12.C.17 Low_Limit

This optional property, of type Double, shall specify a limit below which the Present_Value must fall before an event is generated. This property is required if intrinsic reporting is supported by this object.

12.C.17.1 Conditions for Generating a TO-OFFNORMAL Event

A TO-OFFNORMAL event is generated under these conditions:

(a) the Present_Value must fall below the Low_Limit for a minimum period of time, specified in the Time_Delay property, and
(b) the LowLimitEnable flag must be set in the Limit_Enable property, and
(c) the TO-OFFNORMAL flag must be set in the Event_Enable property.

12.C.17.2 Conditions for Generating a TO-NORMAL Event

Once the Present_Value has fallen below the LowLimit, the Present_Value must exceed the Low_Limit plus the Deadband before a TO-NORMAL event is generated under these conditions:

(a) the Present_Value must exceed the Low_Limit plus the Deadband for a minimum period of time, specified in the Time_Delay property, and
(b) the LowLimitEnable flag must be set in the Limit_Enable property, and
(c) the TO-NORMAL flag must be set in the Event_Enable property.

12.C.18 Deadband

This optional property, of type Double, shall specify a range between the High_Limit and Low_Limit properties within which the Present_Value must remain for a TO-NORMAL event to be generated under these conditions:

(a) the Present_Value must fall below the High_Limit minus Deadband, and
(b) the Present_Value must exceed the Low_Limit plus the Deadband, and
(c) the Present_Value must remain within this range for a minimum period of time, specified in the Time_Delay property, and
(d) either the HighLimitEnable or LowLimitEnable flag must be set in the Limit_Enable property, and
(e) the TO-NORMAL flag must be set in the Event_Enable property.

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

12.C.19 Limit_Enable

This optional property, of type BACnetLimitEnable, shall convey two flags that separately enable and disable reporting of HighLimit and LowLimit offnormal events and their return to normal. This property is required if intrinsic reporting is supported by this object.

12.C.20 Event_Enable

This optional 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.C.21 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 that 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 that no acknowledgment is expected).

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

12.C.22 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.C.23 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.C.24 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.

12.D BitString Value Object Type

The BitString Value object type defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a BitString Value object to make any kind of bitstring data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

Table 12-D. Properties of the BitString Value Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	BIT STRING	R[1]
Bit_Text	BACnetARRAY[N] of CharacterString	O
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	O[3]
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	O
Priority_Array	BACnetPriorityArray	O[2]
Relinquish_Default	BIT STRING	O[2]
Time_Delay	Unsigned	O[3]
Notification_Class	Unsigned	O[3]
Alarm_Values	BACnetARRAY[N] of BIT STRING	O[3]
Bit_Mask	BIT STRING	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] This property is required to be writable when Out_Of_Service is TRUE.
[2] If Present_Value is commandable, then both of these properties shall be present.
[3] These properties are required if the object supports intrinsic reporting.

12.D.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.D.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.D.3 Object_Type

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

12.D.4 Description

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

12.D.5 Present_Value

This property, of type BIT STRING, indicates the current value of the object. The Present_Value property shall be writable when Out_Of_Service is TRUE (see Clause 12.D.10).

12.D.6 Bit_Text

This optional property is a BACnetARRAY of Character Strings representing descriptions of all possible bits of the Present_Value. The number of descriptions matches the number of bits in the Present_Value property. The bits in Present_Value have a one-to-one correspondence with one-based indices in the array, where bit (0) corresponds to array index one.

12.D.7 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a BitString Value object. Three of the flags are associated with the values of another property of this object. A more detailed status could be determined by reading the property that is linked to this flag. 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	Always Logical FALSE (0).
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. In this context "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. Otherwise, the value is logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property is present and has a value of TRUE, otherwise logical FALSE (0).

12.D.8 Event_State

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

12.D.9 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the BitString Value object is reliably reporting its value. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, COMMUNICATION_FAILURE, UNRELIABLE_OTHER}

12.D.10 Out_Of_Service

This optional property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present_Value of the BitString Value object is decoupled from software local to the BACnet device in which the object resides that normally

produces the Present_Value as an output or consumes it as an input. When Out_Of_Service is TRUE, the Present_Value property may be written to freely.

12.D.11 Priority_Array

This optional property is a read-only array containing prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism. If either the Priority_Array property or the Relinquish_Default property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.D.12 Relinquish_Default

This optional property, of type BIT STRING, 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. If either the Relinquish_Default property or the Priority_Array property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.D.13 Time_Delay

This optional property, of type Unsigned, shall specify the minimum period of time in seconds during which the Present_Value must either equal, after applying the Bit_Mask, at least one of the values in the Alarm_Values property before a TO-OFFNORMAL event is generated, or be not equal to, after applying the Bit_Mask, 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.D.14 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.D.15 Alarm_Values

This optional property, of type BACnetARRAY of BIT STRING, shall specify any values that the Present_Value must equal, after applying the Bit_Mask, before a TO-OFFNORMAL event is generated. Only the bits indicated by the Bit_Mask are significant. This property is required if intrinsic reporting is supported by this object.

12.D.15.1 Conditions for Generating a TO-OFFNORMAL Event

A TO-OFFNORMAL event is generated under these conditions:

(a) the Present_Value must remain equal to at least one of the values in the Alarm_Values list, after applying the Bit_Mask, for a minimum period of time, specified in the Time_Delay property, and
(b) the TO-OFFNORMAL flag must be enabled in the Event_Enable property.

12.D.15.2 Conditions for Generating a TO-NORMAL Event

Once equal, the Present_Value must become not equal to any of the values of this property, after applying Bit_Mask before a TO-NORMAL event is generated under these conditions:

(a) the Present_Value must remain not equal to any of the values in the Alarms_Values list, after applying the Bit_Mask, for a minimum period of time, specified by the Time_Delay property, and
(b) the TO-NORMAL flag must be enabled in the Event_Enable property.

12.D.16 Bit_Mask

This optional property, of type BIT STRING, shall specify a bitmask that is used to indicate which bits in the Present_Value are to be monitored by the intrinsic alarm algorithm. A value of one in a bit position indicates that the bit in this position in the Present_Value is to be monitored by the algorithm. A value of zero in a bit position indicates that the bit in this position in the Present_Value is not significant for the purpose of the algorithm.

12.D.17 Event_Enable

This optional 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.D.18 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 that 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 that no acknowledgment is expected).

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

12.D.19 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.D.20 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.D.21 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.

12.E OctetString Value Object Type

The OctetString Value object type defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use an OctetString Value object to make any kind of OCTET STRING data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

Table 12-E. Properties of the OctetString Value Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	OCTET STRING	R[1]
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	O
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	O
Priority_Array	BACnetPriorityArray	O[2]
Relinquish_Default	OCTET STRING	O[2]
Profile_Name	CharacterString	O

[1] This property is required to be writable when Out_Of_Service is TRUE.
[2] If Present_Value is commandable, then both of these properties shall be present.

12.E.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.E.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.E.3 Object_Type

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

12.E.4 Description

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

12.E.5 Present_Value

This property, of type OCTET STRING, indicates the current value of the object. The Present_Value property shall be writable when Out_Of_Service is TRUE (see Clause 12.E.9).

12.E.6 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of an OctetString Value object. Three of the flags are associated with the values of another property of this object. A more detailed status could be determined by reading the property that is linked to this flag. 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 TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).
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. In this context "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. Otherwise, the value is logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property is present and has a value of TRUE, otherwise logical FALSE (0).

12.E.7 Event_State

This optional 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 Reliability property is not present, or if the Reliability property is present and has a value of NO_FAULT_DETECTED, 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.

12.E.8 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the OctetString Value object is reliably reporting its value. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, COMMUNICATION_FAILURE, UNRELIABLE_OTHER}

12.E.9 Out_Of_Service

This optional property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present_Value of the OctetString Value object is decoupled from software local to the BACnet device in which the object resides that normally produces the Present_Value as an output or consumes it as an input. When Out_Of_Service is TRUE, the Present_Value property may be written to freely.

12.E.10 Priority_Array

This optional property, of type BACnetPriorityArray, is a read-only array containing prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism. If either the Priority_Array property or the Relinquish_Default property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.E.11 Relinquish_Default

This optional property, of type OCTET STRING, 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. If either the Relinquish_Default property or the Priority_Array property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.E.12 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.

12.F Time Value Object Type

The Time Value object type defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a Time Value object to make any kind of time data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

A Time Value object is used to represent a single moment in time. In contrast, the Time Pattern Value object can be used to represent multiple recurring times.

Table 12-F. Properties of the Time Value Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	Time	R[1]
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	O
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	O
Priority_Array	BACnetPriorityArray	O[2]
Relinquish_Default	Time	O[2]
Profile_Name	CharacterString	O

[1] If Present_Value is commandable, then it is required to be writable. This property is required to be writable when Out_Of_Service is TRUE.
[2] If Present_Value is commandable, then both of these properties shall be present.

12.F.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.F.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.F.3 Object_Type

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

12.F.4 Description

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

12.F.5 Present_Value

This property, of type Time, indicates the current value of the object. The value of this property shall contain either a fully specified time of day or it shall indicate a fully unspecified time by setting all octets to X'FF'. A fully specified time

shall contain no octets that are equal to X'FF'. This property shall always be used to indicate a time of day, not a duration or relative time value. The Present_Value property shall be writable when Out_Of_Service is TRUE (see Clause 12.F.9).

12.F.6 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a Time Value object. Three of the flags are associated with the values of another property of this object. A more detailed status could be determined by reading the property that is linked to this flag. 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 TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).
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. In this context "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. Otherwise, the value is logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property is present and has a value of TRUE, otherwise logical FALSE (0).

12.F.7 Event_State

This optional 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 Reliability property is not present, or if the Reliability property is present and has a value of NO_FAULT_DETECTED, 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.

12.F.8 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the Time Value object is reliably reporting its value. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, COMMUNICATION_FAILURE, UNRELIABLE_OTHER}

12.F.9 Out_Of_Service

This optional property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present_Value of the Time Value object is decoupled from software local to the BACnet device in which the object resides that normally produces the Present_Value as an output or consumes it as an input. When Out_Of_Service is TRUE, the Present_Value property may be written to freely.

12.F.10 Priority_Array

This optional property, of type BACnetPriorityArray, is a read-only array containing prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism. If either the Priority_Array

property or the Relinquish_Default property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.F.11 Relinquish_Default

This optional property, of type Time, 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. If either the Relinquish_Default property or the Priority_Array property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.F.12 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 is not required to 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.

12.G Integer Value Object Type

The Integer Value object type defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use an Integer Value object to make any kind of signed integer data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

Table 12-G. Properties of the Integer Value Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	INTEGER	R[1]
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	O[4]
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	O
Units	BACnetEngineeringUnits	R
Priority_Array	BACnetPriorityArray	O[2]
Relinquish_Default	INTEGER	O[2]
COV_Increment	Unsigned	O[3]
Time_Delay	Unsigned	O[4]
Notification_Class	Unsigned	O[4]
High_Limit	INTEGER	O[4]
Low_Limit	INTEGER	O[4]
Deadband	Unsigned	O[4]
Limit_Enable	BACnetLimitEnable	O[4]
Event_Enable	BACnetEventTransitionBits	O[4]
Acked_Transitions	BACnetEventTransitionBits	O[4]
Notify_Type	BACnetNotifyType	O[4]
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	O[4]
Profile_Name	CharacterString	O

[1] If Present_Value is commandable, then it is required to be writable. This property is required to be writable when Out_Of_Service is TRUE.
[2] If Present_Value is commandable, then both of these properties shall be present.
[3] This property is required if the object supports COV reporting.
[4] These properties are required if the object supports intrinsic reporting.

12.G.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.G.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.G.3 Object_Type

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

12.G.4 Description

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

12.G.5 Present_Value

This property, of type INTEGER, indicates the current value of the object. The Present_Value property shall be writable when Out_Of_Service is TRUE (see Clause 12.G.9).

12.G.6 Status_Flags

This required property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of an Integer Value object. Three of the flags are associated with the values of another property of this object. A more detailed status could be determined by reading the property that is linked to this flag. 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 TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).
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. In this context "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. Otherwise, the value is logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property is present and has a value of TRUE, otherwise logical FALSE (0).

12.G.7 Event_State

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

12.G.8 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the Integer Value object is reliably reporting its value. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, COMMUNICATION_FAILURE, UNRELIABLE_OTHER}

12.G.9 Out_Of_Service

This optional property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present_Value of the Integer Value object is decoupled from software local to the BACnet device in which the object resides that normally produces the Present_Value as an output or consumes it as an input. When Out_Of_Service is TRUE, the Present_Value property may be written to freely.

12.G.10 Units

This property, of type BACnetEngineeringUnits, indicates the measurement units of this object. See the BACnetEngineeringUnits ASN.1 production in Clause 21 for a list of engineering units defined by this standard.

12.G.11 Priority_Array

This optional property, of type BACnetPriorityArray, is a read-only array containing prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism. If either the Priority_Array property or the Relinquish_Default property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.G.12 Relinquish_Default

This optional property, of type INTEGER, 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. If either the Relinquish_Default property or the Priority_Array property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.G.13 COV_Increment

This optional property, of type Unsigned, shall specify the minimum change in Present_Value that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

12.G.14 Time_Delay

This optional property, of type Unsigned, shall specify the minimum period of time in seconds during which the Present_Value must remain outside the band defined by the High_Limit and Low_Limit properties before a TO-OFFNORMAL event is generated, or within the same band, including the Deadband property, before a TO-NORMAL event is generated. This property is required if intrinsic reporting is supported by this object.

12.G.15 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.G.16 High_Limit

This optional property, of type INTEGER, shall specify a limit that the Present_Value must exceed before an event is generated. This property is required if intrinsic reporting is supported by this object.

12.G.16.1 Conditions for Generating a TO-OFFNORMAL Event

A TO-OFFNORMAL event is generated under these conditions:

(a) the Present_Value must exceed the High_Limit for a minimum period of time, specified in the Time_Delay property, and
(b) the HighLimitEnable flag must be set in the Limit_Enable property, and
(c) the TO-OFFNORMAL flag must be set in the Event_Enable property.

12.G.16.2 Conditions for Generating a TO-NORMAL Event

Once exceeded, the Present_Value must fall below the High_Limit minus the Deadband before a TO-NORMAL event is generated under these conditions:

(a) the Present_Value must fall below the High_Limit minus the Deadband for a minimum period of time, specified in the Time_Delay property, and
(b) the HighLimitEnable flag must be set in the Limit_Enable property, and
(c) the TO-NORMAL flag must be set in the Event_Enable property.

12.G.17 Low_Limit

This optional property, of type INTEGER, shall specify a limit below which the Present_Value must fall before an event is generated. This property is required if intrinsic reporting is supported by this object.

12.G.17.1 Conditions for Generating a TO-OFFNORMAL Event

A TO-OFFNORMAL event is generated under these conditions:

(a) the Present_Value must fall below the Low_Limit for a minimum period of time, specified in the Time_Delay property, and
(b) the LowLimitEnable flag must be set in the Limit_Enable property, and
(c) the TO-OFFNORMAL flag must be set in the Event_Enable property.

12.G.17.2 Conditions for Generating a TO-NORMAL Event

Once the Present_Value has fallen below the LowLimit, the Present_Value must exceed the Low_Limit plus the Deadband before a TO-NORMAL event is generated under these conditions:

(a) the Present_Value must exceed the Low_Limit plus the Deadband for a minimum period of time, specified in the Time_Delay property, and
(b) the LowLimitEnable flag must be set in the Limit_Enable property, and
(c) the TO-NORMAL flag must be set in the Event_Enable property.

12.G.18 Deadband

This optional property, of type Unsigned, shall specify a range between the High_Limit and Low_Limit properties within which the Present_Value must remain for a TO-NORMAL event to be generated under these conditions:

(a) the Present_Value must fall below the High_Limit minus Deadband, and
(b) the Present_Value must exceed the Low_Limit plus the Deadband, and
(c) the Present_Value must remain within this range for a minimum period of time, specified in the Time_Delay property, and
(d) either the HighLimitEnable or LowLimitEnable flag must be set in the Limit_Enable property, and
(e) the TO-NORMAL flag must be set in the Event_Enable property.

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

12.G.19 Limit_Enable

This optional property, of type BACnetLimitEnable, shall convey two flags that separately enable and disable reporting of HighLimit and LowLimit offnormal events and their return to normal. This property is required if intrinsic reporting is supported by this object.

12.G.20 Event_Enable

This optional 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.G.21 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 that 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 that no acknowledgment is expected).

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

12.G.22 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.G.23 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.G.24 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.

12.H Positive Integer Value Object Type

The Positive Integer Value object type defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a Positive Integer Value object to make any kind of unsigned data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

Table 12-H. Properties of the Positive Integer Value Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	Unsigned	R[1]
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	O[4]
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	O
Units	BACnetEngineeringUnits	R
Priority_Array	BACnetPriorityArray	O[2]
Relinquish_Default	Unsigned	O[2]
COV_Increment	Unsigned	O[3]
Time_Delay	Unsigned	O[4]
Notification_Class	Unsigned	O[4]
High_Limit	Unsigned	O[4]
Low_Limit	Unsigned	O[4]
Deadband	Unsigned	O[4]
Limit_Enable	BACnetLimitEnable	O[4]
Event_Enable	BACnetEventTransitionBits	O[4]
Acked_Transitions	BACnetEventTransitionBits	O[4]
Notify_Type	BACnetNotifyType	O[4]
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	O[4]
Profile_Name	CharacterString	O

[1] If Present_Value is commandable, then it is required to be writable. This property is required to be writable when Out_Of_Service is TRUE.
[2] If Present_Value is commandable, then both of these properties shall be present.
[3] This property is required if the object supports COV reporting.
[4] These properties are required if the object supports intrinsic reporting.

12.H.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.H.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.H.3 Object_Type

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

12.H.4 Description

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

12.H.5 Present_Value

This property, of type Unsigned, indicates the current value of the object. The Present_Value property shall be writable when Out_Of_Service is TRUE (see Clause 12.H.9).

12.H.6 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a Positive Integer Value object. Three of the flags are associated with the values of another property of this object. A more detailed status could be determined by reading the property that is linked to this flag. 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 TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).
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. In this context "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. Otherwise, the value is logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property is present and has a value of TRUE, otherwise logical FALSE (0).

12.H.7 Event_State

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

12.H.8 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the Positive Integer Value object is reliably reporting its value. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, COMMUNICATION_FAILURE, UNRELIABLE_OTHER}

12.H.9 Out_Of_Service

This optional property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present_Value of the Positive Integer Value object is decoupled from software local to the BACnet device in which the object resides that normally produces the Present_Value as an output or consumes it as an input. When Out_Of_Service is TRUE, the Present_Value property may be written to freely.

12.H.10 Units

This property, of type BACnetEngineeringUnits, indicates the measurement units of this object. See the BACnetEngineeringUnits ASN.1 production in Clause 21 for a list of engineering units defined by this standard.

12.H.11 Priority_Array

This optional property, of type BACnetPriorityArray, is a read-only array containing prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism. If either the Priority_Array property or the Relinquish_Default property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.H.12 Relinquish_Default

This optional property, of type Unsinged, 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. If either the Relinquish_Default property or the Priority_Array property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.H.13 COV_Increment

This optional property, of type Unsigned, shall specify the minimum change in Present_Value that will cause a COVNotification to be issued to subscriber COV-clients. This property is required if COV reporting is supported by this object.

12.H.14 Time_Delay

This optional property, of type Unsigned, shall specify the minimum period of time in seconds during which the Present_Value must remain outside the band defined by the High_Limit and Low_Limit properties before a TO-OFFNORMAL event is generated, or within the same band, including the Deadband property, before a TO-NORMAL event is generated. This property is required if intrinsic reporting is supported by this object.

12.H.15 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.H.16 High_Limit

This optional property, of type Unsigned, shall specify a limit that the Present_Value must exceed before an event is generated. This property is required if intrinsic reporting is supported by this object.

12.H.16.1 Conditions for Generating a TO-OFFNORMAL Event

A TO-OFFNORMAL event is generated under these conditions:

(a) the Present_Value must exceed the High_Limit for a minimum period of time, specified in the Time_Delay property, and
(b) the HighLimitEnable flag must be set in the Limit_Enable property, and
(c) the TO-OFFNORMAL flag must be set in the Event_Enable property.

12.H.16.2 Conditions for Generating a TO-NORMAL Event

Once exceeded, the Present_Value must fall below the High_Limit minus the Deadband before a TO-NORMAL event is generated under these conditions:

(a) the Present_Value must fall below the High_Limit minus the Deadband for a minimum period of time, specified in the Time_Delay property, and
(b) the HighLimitEnable flag must be set in the Limit_Enable property, and
(c) the TO-NORMAL flag must be set in the Event_Enable property.

12.H.17 Low_Limit

This optional property, of type Unsigned, shall specify a limit below which the Present_Value must fall before an event is generated. This property is required if intrinsic reporting is supported by this object.

12.H.17.1 Conditions for Generating a TO-OFFNORMAL Event

A TO-OFFNORMAL event is generated under these conditions:

(a) the Present_Value must fall below the Low_Limit for a minimum period of time, specified in the Time_Delay property, and
(b) the LowLimitEnable flag must be set in the Limit_Enable property, and
(c) the TO-OFFNORMAL flag must be set in the Event_Enable property.

12.H.17.2 Conditions for Generating a TO-NORMAL Event

Once the Present_Value has fallen below the LowLimit, the Present_Value must exceed the Low_Limit plus the Deadband before a TO-NORMAL event is generated under these conditions:

(a) the Present_Value must exceed the Low_Limit plus the Deadband for a minimum period of time, specified in the Time_Delay property, and
(b) the LowLimitEnable flag must be set in the Limit_Enable property, and
(c) the TO-NORMAL flag must be set in the Event_Enable property.

12.H.18 Deadband

This optional property, of type Unsigned, shall specify a range between the High_Limit and Low_Limit properties within which the Present_Value must remain for a TO-NORMAL event to be generated under these conditions:

(a) the Present_Value must fall below the High_Limit minus Deadband, and
(b) the Present_Value must exceed the Low_Limit plus the Deadband, and
(c) the Present_Value must remain within this range for a minimum period of time, specified in the Time_Delay property, and
(d) either the HighLimitEnable or LowLimitEnable flag must be set in the Limit_Enable property, and
(e) the TO-NORMAL flag must be set in the Event_Enable property.

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

12.H.19 Limit_Enable

This optional property, of type BACnetLimitEnable, shall convey two flags that separately enable and disable reporting of HighLimit and LowLimit offnormal events and their return to normal. This property is required if intrinsic reporting is supported by this object.

12.H.20 Event_Enable

This optional 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.H.21 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 that 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 that no acknowledgment is expected).

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

12.H.22 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.H.23 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.H.24 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.

12.I Date Value Object Type

The Date Value object type defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a Date Value object to make any kind of date data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

A Date Value object is used to represent a single day. In contrast, the Date Pattern Value object can be used to represent multiple recurring dates.

Table 12-I. Properties of the Date Value Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	Date	R[1]
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	O
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	O
Priority_Array	BACnetPriorityArray	O[2]
Relinquish_Default	Date	O[2]
Profile_Name	CharacterString	O

[1] If Present_Value is commandable, then it is required to be writable. This property is required to be writable when Out_Of_Service is TRUE.

[2] If Present_Value is commandable, then both of these properties shall be present.

12.I.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.I.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.I.3 Object_Type

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

12.I.4 Description

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

12.I.5 Present_Value

This property, of type Date, indicates the current value of the object. The value of this property shall contain either a fully specified date or it shall indicate a fully unspecified date by setting all octets to X'FF'. A fully specified date shall

not contain octets that are equal to X'FF' or contain special values for the 'month' or 'day of month' fields. The Present_Value property shall be writable when Out_Of_Service is TRUE (see Clause 12.I.9).

12.I.6 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a Date Value object. Three of the flags are associated with the values of another property of this object. A more detailed status could be determined by reading the property that is linked to this flag. 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 TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).
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. In this context "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. Otherwise, the value is logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property is present and has a value of TRUE, otherwise logical FALSE (0).

12.I.7 Event_State

This optional 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 Reliability property is not present, or if the Reliability property is present and has a value of NO_FAULT_DETECTED, 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.

12.I.8 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the Date Value object is reliably reporting its value. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, COMMUNICATION_FAILURE, UNRELIABLE_OTHER}

12.I.9 Out_Of_Service

This optional property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present_Value of the Date Value object is decoupled from software local to the BACnet device in which the object resides that normally produces the Present_Value as an output or consumes it as an input. When Out_Of_Service is TRUE, the Present_Value property may be written to freely.

12.I.10 Priority_Array

This optional property, of type BACnetPriorityArray, is a read-only array containing prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism. If either the Priority_Array property or the Relinquish_Default property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.I.11 Relinquish_Default

This optional property, of type Date, 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. If either the Relinquish_Default property or the Priority_Array property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.I.12 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.

12.J DateTime Pattern Value Object Type

The DateTime Pattern Value object type defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a DateTime Pattern Value object to make any kind of datetime data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

DateTime Pattern objects can be used to represent multiple recurring dates and times based on rules defined by the pattern of individual fields of the date and time, some of which can be special values like "even months", or "don't care", which matches any value in that field. Examples of possibilities would be: "11:00 every Thursday in any June", or "every day in May 2009".

Table 12-J. Properties of the DateTime Pattern Value Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	BACnetDateTime	R[1]
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	O
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	O
Is_UTC	BOOLEAN	O
Priority_Array	BACnetPriorityArray	O[2]
Relinquish_Default	BACnetDateTime	O[2]
Profile_Name	CharacterString	O

[1] If Present_Value is commandable, then it is required to be writable. This property is required to be writable when Out_Of_Service is TRUE.
[2] If Present_Value is commandable, then both of these properties shall be present.

12.J.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.J.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.J.3 Object_Type

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

12.J.4 Description

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

12.J.5 Present_Value

This property, of type BACnetDateTime, indicates the current value of the object. The value of this property may indicate a fully specified date and time or a partially specified datetime pattern by containing one or more unspecified octets that are equal to X'FF' or the special values for the 'month' or 'day of month' fields. The Present_Value property shall be writable when Out_Of_Service is TRUE (see Clause 12.J.9).

12.J.6 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a DateTime Value object. Three of the flags are associated with the values of another property of this object. A more detailed status could be determined by reading the property that is linked to this flag. 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 TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).
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. In this context "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. Otherwise, the value is logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property is present and has a value of TRUE, otherwise logical FALSE (0).

12.J.7 Event_State

This optional 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 Reliability property is not present, or if the Reliability property is present and has a value of NO_FAULT_DETECTED, 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.

12.J.8 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the DateTime Value object is reliably reporting its value. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, COMMUNICATION_FAILURE, UNRELIABLE_OTHER}

12.J.9 Out_Of_Service

This optional property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present_Value of the DateTime Value object is decoupled from software local to the BACnet device in which the object resides that normally produces the Present_Value as an output or consumes it as an input. When Out_Of_Service is TRUE, the Present_Value property may be written to freely.

12.J.10 Priority_Array

This optional property, of type BACnetPriorityArray, is a read-only array containing prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism. If either the Priority_Array property or the Relinquish_Default property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.J.11 Relinquish_Default

This optional property, of type Date, 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. If either the Relinquish_Default property or the Priority_Array property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.J.12 Is_UTC

This optional property indicates whether the Present_Value property indicates a UTC date and time (when TRUE) or a local date and time (when FALSE). If this property is absent, the Present_Value shall be a local date and time.

12.J.13 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.

12.K Time Pattern Value Object Type

The Time Pattern Value object type defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a Time Pattern Value object to make any kind of time data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

Time Pattern objects can be used to represent multiple recurring times based on rules definted by the pattern of individual fields of the time, some of which may be "don't care", which matches any value in that field. Examples of possibilities would be: "every minute of the 11 o'clock hour of the day", or "the thirteenth minute of any hour".

Table 12-K. Properties of the Time Pattern Value Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	Time	R[1]
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	O
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	O
Priority_Array	BACnetPriorityArray	O[2]
Relinquish_Default	Time	O[2]
Profile_Name	CharacterString	O

[1] If Present_Value is commandable, then it is required to be writable. This property is required to be writable when Out_Of_Service is TRUE.
[2] If Present_Value is commandable, then both of these properties shall be present.

12.K.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.K.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.K.3 Object_Type

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

12.K.4 Description

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

12.K.5 Present_Value

This property, of type Time, indicates the current value of the object. The value of this property may indicate a fully specified time or a partially specified time pattern by containing one or more "unspecified" octets that are equal to X'FF'. The Present_Value property shall be writable when Out_Of_Service is TRUE (see Clause 12.K.9).

12.K.6 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a Time Value object. Three of the flags are associated with the values of another property of this object. A more detailed status could be determined by reading the property that is linked to this flag. 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 TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).
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. In this context "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. Otherwise, the value is logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property is present and has a value of TRUE, otherwise logical FALSE (0).

12.K.7 Event_State

This optional 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 Reliability property is not present, or if the Reliability property is present and has a value of NO_FAULT_DETECTED, 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.

12.K.8 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the Time Value object is reliably reporting its value. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, COMMUNICATION_FAILURE, UNRELIABLE_OTHER}

12.K.9 Out_Of_Service

This optional property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present_Value of the Time Value object is decoupled from software local to the BACnet device in which the object resides that normally produces the Present_Value as an output or consumes it as an input. When Out_Of_Service is TRUE, the Present_Value property may be written to freely.

12.K.10 Priority_Array

This optional property, of BACnetPriorityArray, is a read-only array containing prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism. If either the Priority_Array property or the Relinquish_Default property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.K.11 Relinquish_Default

This optional property, of type Time, 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. If either the Relinquish_Default property or the Priority_Array property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.K.12 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 is not required to 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.

12.L Date Pattern Value Object Type

The Date Pattern Value object type defines a standardized object whose properties represent the externally visible characteristics of a named data value in a BACnet device. A BACnet device can use a Date Pattern Value object to make any kind of date data value accessible to other BACnet devices. The mechanisms by which the value is derived are not visible to the BACnet client.

Date Pattern objects can be used to represent multiple recurring dates and times based on rules definted by the pattern of individual fields of the date and time, some of which can be special values like "even montes", or "don't care", which matches any value in that field. Examples of possibilities would be: "every Thursday in May of any year", or "every day in May 2009".

Table 12-L. Properties of the Date Pattern Value Object

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Present_Value	Date	R[1]
Status_Flags	BACnetStatusFlags	R
Event_State	BACnetEventState	O
Reliability	BACnetReliability	O
Out_Of_Service	BOOLEAN	O
Priority_Array	BACnetPriorityArray	O[2]
Relinquish_Default	Date	O[2]
Profile_Name	CharacterString	O

[1] If Present_Value is commandable, then it is required to be writable. This property is required to be writable when Out_Of_Service is TRUE.
[2] If Present_Value is commandable, then both of these properties shall be present.

12.L.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.L.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.L.3 Object_Type

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

12.L.4 Description

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

12.L.5 Present_Value

This property, of type Date, indicates the current value of the object. The value of this property may indicate a fully specified date or a partially specified date pattern by containing one or more "unspecified" octets that are equal to X'FF' or the special values for the 'month' or 'day of month' fields. The Present_Value property shall be writable when Out_Of_Service is TRUE (see Clause 12.L.9).

12.L.6 Status_Flags

This property, of type BACnetStatusFlags, represents four Boolean flags that indicate the general "health" of a Date Value object. Three of the flags are associated with the values of another property of this object. A more detailed status could be determined by reading the property that is linked to this flag. 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 TRUE (1) if the Event_State property is present and does not have a value of NORMAL, otherwise logical FALSE (0).
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. In this context "overridden" is taken to mean that the Present_Value property is not changeable through BACnet services. Otherwise, the value is logical FALSE (0).
OUT_OF_SERVICE	Logical TRUE (1) if the Out_Of_Service property is present and has a value of TRUE, otherwise logical FALSE (0).

12.L.7 Event_State

This optional 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 Reliability property is not present, or if the Reliability property is present and has a value of NO_FAULT_DETECTED, 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.

12.L.8 Reliability

This optional property, of type BACnetReliability, provides an indication of whether the Date Value object is reliably reporting its value. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, COMMUNICATION_FAILURE, UNRELIABLE_OTHER}

12.L.9 Out_Of_Service

This optional property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the Present_Value of the Date Value object is decoupled from software local to the BACnet device in which the object resides that normally produces the Present_Value as an output or consumes it as an input. When Out_Of_Service is TRUE, the Present_Value property may be written to freely.

12.L.10 Priority_Array

This optional property, of type BACnetPriorityArray, is a read-only array containing prioritized commands that are in effect for this object. See Clause 19 for a description of the prioritization mechanism. If either the Priority_Array property or the Relinquish_Default property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.L.11 Relinquish_Default

This optional property, of type Date, 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. If either the Relinquish_Default property or the Priority_Array property is present, then both of them shall be present. If Present_Value is commandable, then Priority_Array and Relinquish_Default shall both be present.

12.L.12 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.

[Change Clause **12.12.5**, p. 188]

12.12.5 Event_Type

This property, of type BACnetEventType, indicates the type of event algorithm that is to be used to detect the occurrence of events and report to enrolled devices. This parameter is an enumerated type that may have any of the following values:

{CHANGE_OF_BITSTRING, …, *DOUBLE_OUT_OF_RANGE, SIGNED_OUT_OF_RANGE, UNSIGNED_OUT_OF_RANGE, CHANGE_OF_CHARACTERSTRING*}.

[Change **Table 12-15**, p.189]

Table 12-15. Event_Types, Event_States, and their Parameters

Event_Type	Event_State	Event_Parameters
CHANGE_OF_BITSTRING	NORMAL OFFNORMAL	Time_Delay Bitmask List_Of_Bitstring_Values
CHANGE_OF_STATE	NORMAL OFFNORMAL	Time_Delay List_Of_Values
CHANGE_OF_VALUE	NORMAL	Time_Delay Bitmask Referenced_Property_Increment
COMMAND_FAILURE	NORMAL OFFNORMAL	Time_Delay Feedback_Property_Reference
FLOATING_LIMIT	NORMAL HIGH_LIMIT LOW_LIMIT	Time_Delay Setpoint_Reference Low_Diff_Limit High_Diff_Limit Deadband
OUT_OF_RANGE, *DOUBLE_OUT_OF_RANGE* *UNSIGNED_OUT_OF_RANGE,* *SIGNED_OUT_OF_RANGE*	NORMAL HIGH_LIMIT LOW_LIMIT	Time_Delay Low_Limit High_Limit Deadband
BUFFER_READY	NORMAL	Notification_Threshold
CHANGE_OF_LIFE_SAFETY	NORMAL OFFNORMAL LIFE_SAFETY_ALARM	Time_Delay List_Of_Alarm_Values List_Of_Life_Safety_Alarm_Values Mode_Property_Reference
CHANGE_OF_CHARACTERSTRING	*NORMAL* *OFFNORMAL*	*Time_Delay* *List_of_Alarm_Values*
EXTENDED	Any BACnetEventState	Vendor_Id Extended_Event_Type Parameters
UNSIGNED_RANGE	NORMAL HIGH_LIMIT LOW_LIMIT	Time_Delay Low_Limit High_Limit

[Change Clause 13.1, p. 290]

13.1 Change of Value Reporting

…

When a BACnet standard object, of a type listed in Table 13-1, supports COV reporting it shall support COV reporting for the property as listed in Table 13-1. At the implementor's discretion, COV reporting may also be supported for any other property of the object. For properties listed in Table 13-1 that have a ~~REAL~~ *numeric* datatype, the COV increment used to determine when to generate notifications will be the COV_Increment property of the object unless a COV_Increment parameter is supplied in the SubscribeCOVProperty service. For other properties that have a ~~REAL~~ *numeric* datatype, the COV increment to use when not supplied with the SubscribeCOVProperty service shall be a local matter. This is to allow multiple subscribers that do not require a specific increment to use a common increment to allow for the reduction of the processing burden on the COV-server. The criteria for COV reporting for properties other than those listed in Table 13-1 is based on the datatype of the property subscribed to and is described in Table 13-1a.

[Change Clause 13.15.1.7 p. 328

13.15.1.7 COV Increment

…

This parameter, of type REAL, shall specify the minimum change in the monitored property that will cause a COVNotification to be issued to subscriber COV-clients. This parameter is ignored if the datatype of the monitored property is not ~~REAL~~*numeric*. If the monitored property is Present_Value, its datatype is ~~REAL~~numeric, this parameter is not present, and the monitored object has a COV_Increment property, then the COV increment to use is taken from the COV_Increment property of the monitored object. Otherwise, the COV increment is a local matter. The intent is to allow the subscriber to use a previously established COV increment from another subscription or to allow use of the COV_Increment property in the monitored object.

[Change **Table 13-1**, p. 291]

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

Object Type	Criteria	Properties Reported
Access Door	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)	Present_Value, Status_Flags, Door_Alarm_State (if the object has a Door_Alarm_State property)
Analog Input, Analog Output, Analog Value Large Analog Value Integer Value Positive Integer Value	If Present_Value changes by COV_Increment or Status_Flags changes at all	Present_Value, Status_Flags
Binary Input, Binary Output, Binary Value, Life Safety Point, Life Safety Zone, Multi-state Input, Multi-state Output, Multi-state Value, *OctetString Value,* *CharacterString Value,* *Time Value,* *DateTime Value,* *Date Value,* *Time Pattern Value,* *Date Pattern Value,* *DateTime Pattern Value*	If Present_Value changes at all or Status_Flags changes at all	Present_Value, Status_Flags
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
Loop	If Present_Value changes by COV_Increment or Status_Flags changes at all	Present_Value, Status_Flags, Setpoint, Controlled_Variable_Value
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

[Add entries to **Table 13-2**, p. 293]

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

Object Type	Criteria	Event Type
…	…	…
CharacterString Value	*If any one of the strings in Alarm_Values appears in any position within the Present_Value (per match criteria 12.A.12) for longer than Time_Delay AND the new transition is enabled in Event_Enable*	*CHANGE_OF_CHARACTERSTRING*
Large Analog Value	*If Present_Value exceeds the range between High_Limit and Low_Limit for longer than Time_Delay AND the new transition is enabled in Event_Enable and Limit_Enable,* *OR* *Present_Value returns within the High_Limit - Deadband to Low_Limit + Deadband range for longer than Time_Delay AND the new transition is enabled in Event_Enable and Limit_Enable*	*DOUBLE_OUT_OF_RANGE*
BitString Value	*If Present_Value, after application of Bit_Mask, equals at least one of the values of the Alarm_Values set for longer than Time_Delay AND the new transition is enabled in Event_Enable*	*CHANGE_OF_BITSTRING*
Integer Value	*If Present_Value exceeds range between High_Limit and Low_Limit for longer than Time_Delay AND the new transition is enabled in Event_Enable and Limit_Enable,* *OR* *Present_Value returns within the High_Limit - Deadband to Low_Limit + Deadband range for longer than Time_Delay AND the new transition is enabled in Event_Enable and Limit_Enable*	*SIGNED_OUT_OF_RANGE*
Positive Integer Value	*If Present_Value exceeds range between High_Limit and Low_Limit for longer than Time_Delay AND the new transition is enabled in Event_Enable and Limit_Enable,* *OR* *Present_Value returns within the High_Limit - Deadband to Low_Limit + Deadband range for longer than Time_Delay AND the new transition is enabled in Event_Enable and Limit_Enable*	*UNSIGNED_OUT_OF_RANGE*

[Add entries to **Table 13-3**, p. 294]

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

Object	Event Type	Notification Parameters	Referenced Object's Properties
…	…	…	…
CharacterString Value	*CHANGE_OF_CHARACTERSTRING*	*Changed_Value* *Status_Flags* *Alarm_Value*	*Present_Value* *Status_Flags* *Alarm_Values[N]*
Large Analog Value	*DOUBLE_OUT_OF_RANGE*	*Exceeded_Value* *Status_Flags* *Deadband* *Exceeding_Limit*	*Present_Value* *Status_Flags* *Deadband* *High_Limit or Low_Limit*
BitString Value	*CHANGE_OF_BITSTRING*	*Referenced_Bitstring* *Status_Flags*	*Present_Value* *Status_Flags*
Integer Value	*SIGNED_OUT_OF_RANGE*	*Exceeding_Value* *Status_Flags* *Deadband* *Exceeding_Limit*	*Present_Value* *Status_Flags* *Deadband* *High_Limit or Low_Limit*
Positive Integer Value	*UNSIGNED_OUT_OF_RANGE*	*Exceeding_Value* *Status_Flags* *Deadband* *Exceeding_Limit*	*Present_Value* *Status_Flags* *Deadband* *High_Limit or Low_Limit*

[1] This parameter conveys a reference to the Log_Buffer property of the object.

[Add entries to **Table 13-4**, p. 295]

Table 13-4. Notification Parameters for Standard Event Types

Event Type	Notification Parameters	Description
…	…	…
CHANGE_OF_CHARACTERSTRING	*Changed_Value* *Status_Flags* *Alarm_Value*	*The new value that contained the alarm.* *The Status_Flags of the referenced object.* *The Alarm_Values string that triggered the alarm.*
DOUBLE_OUT_OF_RANGE	*Exceeding_Value* *Status_Flags* *Deadband* *Exceeding_Limit*	*The value that exceeded a limit.* *The Status_Flags of the referenced object.* *The deadband used for limit checking.* *The limit that was exceeded.*
SIGNED_OUT_OF_RANGE	*Exceeding_Value* *Status_Flags* *Deadband* *Exceeding_Limit*	*The value that exceeded a limit.* *The Status_Flags of the referenced object.* *The deadband used for limit checking.* *The limit that was exceeded.*
UNSIGNED_OUT_OF_RANGE	*Exceeding_Value* *Status_Flags* *Deadband* *Exceeding_Limit*	*The value that exceeded a limit.* *The Status_Flags of the referenced object.* *The deadband used for limit checking.* *The limit that was exceeded.*

[Change Clause 13.3, p. 296]

13.3 Algorithmic Change Reporting

…

The following event type algorithms are specified in this standard because of their widespread occurrence in building automation and control systems. They are:

(a) CHANGE_OF_BITSTRING
(b) CHANGE_OF_STATE
(c) CHANGE_OF_VALUE
(d) COMMAND_FAILURE
(e) FLOATING_LIMIT
(f) OUT_OF_RANGE
(g) BUFFER_READY
(h) CHANGE_OF_LIFE_SAFETY
(i) UNSIGNED_RANGE
(j) DOUBLE_OUT_OF_RANGE
(k) SIGNED_OUT_OF_RANGE
(l) UNSIGNED_OUT_OF_RANGE
(m) CHANGE_OF_CHARACTERSTRING

[Change title of Figure 13-8, p. 301]

Figure 13-8. OUT_OF_RANGE*, DOUBLE_OUT_OF_RANGE, SIGNED_OUT_OF_RANGE, and UNSIGNED_OUT_OF_RANGE* algorithm.

[Add new Clause **13.3.W**, p.303]

13.3.W CHANGE_OF_CHARACTERSTRING Algorithm

A CHANGE_OF_CHARACTERSTRING occurs when the value of the referenced property contains one of the non-empty values in the List_Of_Values, and continues to contain a non-empty value from the list for Time_Delay seconds. For the purposes of event notification, CHANGE_OF_CHARACTERSTRING events generate a TO-OFFNORMAL transition.

Once in the OFF_NORMAL state, if the referenced value changes to contain a different non-empty value from the List_Of_Values for Time_Delay seconds, the algorithm will generate another TO-OFFNORMAL transition.

A CHANGE_OF_CHARACTERSTRING event clears when the value of the referenced property no longer contains any of the non-empty values in the List_Of_Values, and that condition remains for Time_Delay seconds. The clearing of a CHANGE_OF_CHARACTERSTRING generates a TO-NORMAL transition. See Figure 13-W.

[Add new Clause **13.3.X**, p.303]

13.3.X DOUBLE_OUT_OF_RANGE Algorithm

A DOUBLE_OUT_OF_RANGE occurs if the referenced property leaves a range of values defined by the High_Limit and Low_Limit parameters and remains there for Time_Delay seconds. If the transition is to a value above the High_Limit or below the Low_Limit, the Event Enrollment object generates a TO-OFFNORMAL transition. The event notification shall show an 'Event Type' of DOUBLE_OUT_OF_RANGE.

A DOUBLE_OUT_OF_RANGE clears when the referenced property attains a value greater than the (Low_Limit + Deadband) or a value less than the (High_Limit - Deadband) and remains there for Time_Delay seconds. The Event Enrollment object generates a TO-NORMAL transition. The event notification shall show an 'Event Type' of DOUBLE_OUT_OF_RANGE. See Figure 13-8.

[Add new Clause **13.3.Y**, p.303]

13.3.Y SIGNED_OUT_OF_RANGE Algorithm

An SIGNED_OUT_OF_RANGE occurs if the referenced property leaves a range of values defined by the High_Limit and Low_Limit parameters and remains there for Time_Delay seconds. If the transition is to a value above the High_Limit or below the Low_Limit, the Event Enrollment object generates a TO-OFFNORMAL transition. The event notification shall show an 'Event Type' of SIGNED_OUT_OF_RANGE.

An SIGNED_OUT_OF_RANGE clears when the referenced property attains a value greater than the (Low_Limit + Deadband) or a value less than the (High_Limit - Deadband) and remains there for Time_Delay seconds. Note that the limit values may be Boolean TRUE or FALSE as well as analog values. The Event Enrollment object generates a TO-NORMAL transition. The event notification shall show an 'Event Type' of SIGNED_OUT_OF_RANGE. See Figure 13-8.

[Add new Clause **13.3.Z**, p.303]

13.3.Z UNSIGNED_OUT_OF_RANGE Algorithm

An UNSIGNED_OUT_OF_RANGE occurs if the referenced property leaves a range of values defined by the High_Limit and Low_Limit parameters and remains there for Time_Delay seconds. If the transition is to a value above the High_Limit or below the Low_Limit, the Event Enrollment object generates a TO-OFFNORMAL transition. The event notification shall show an 'Event Type' of UNSIGNED_OUT_OF_RANGE.

An UNSIGNED_OUT_OF_RANGE clears when the referenced property attains a value greater than the (Low_Limit + Deadband) or a value less than the (High_Limit - Deadband) and remains there for Time_Delay seconds. Note that the limit values may be Boolean TRUE or FALSE as well as analog values. The Event Enrollment object generates a TO-NORMAL transition. The event notification shall show an 'Event Type' of UNSIGNED_OUT_OF_RANGE. See Figure 13-8.

[Add new figure, Figure 13-W]

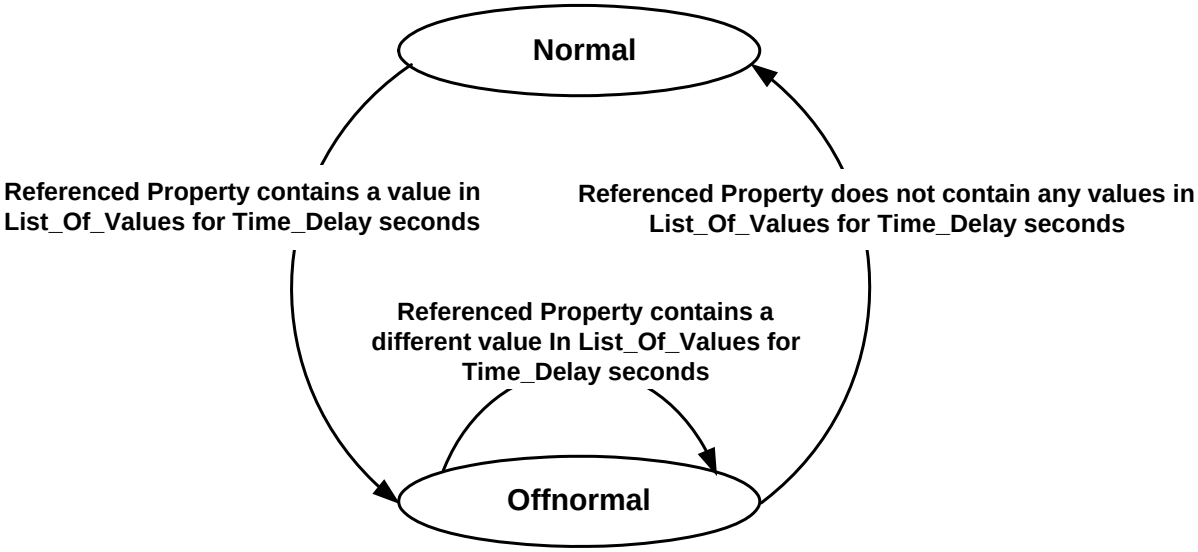

Figure 13-W. CHANGE_OF_CHARACTERSTRING algorithm.

[Insert new entry to **Table** 15-8, p.347]

Table 15-8. Valid Boolean Operators for BACnet Datatypes

Datatype	Operators Allowed	Datatype	Operators Allowed
NULL	= ≠	CharacterString	= ≠ < > ≤ ≥
BOOLEAN	= ≠	OCTET STRING	= ≠
Unsigned	= ≠ < > ≤ ≥	ENUMERATED	= ≠
INTEGER	= ≠ < > ≤ ≥	Date	= ≠ < > ≤ ≥
REAL	= ≠ < > ≤ ≥	Time	= ≠ < > ≤ ≥
BIT STRING	= ≠	*DOUBLE*	= ≠ < > ≤ ≥
BACnetObjectIdentifier	= ≠ < > ≤ ≥	Others	= ≠

[Change Clause 19.2.1.1, p.402]

19.2.1.1 Commandable Properties

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

OBJECT	COMMANDABLE PROPERTY
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
Access Door	Present_Value
BitString Value	*Present_Value*

CharacterString Value	*Present_Value*
Date Value	*Present_Value*
Date Pattern Value	*Present_Value*
DateTime Value	*Presetn_Value*
DateTime Pattern Value	*Present_Value*
Large Analog Value	*Present_Value*
OctetString Value	*Present_Value*
Integer Value	*Present_Value*
Time Value	*Present_Value*
Time Patterm Value	*Present_Value*
Positive Integer Value	*Present_Value*

The designated properties of the Analog Output, Binary Output, 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, *BitString Value, CharacterString Value, Date Value, Date Pattern Value, DateTime Value, DateTime Pattern Value, Large Analog Value, OctetString Value, Integer Value, Time Value, Time Patterm Value, and Positive Integer 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.

[Change **BACnetCOVSubscription** production in Clause **21**, p.449]

```
BACnetCOVSubscription ::= SEQUENCE {
        recipient                   [0] BACnetRecipientProcess,
        monitoredPropertyReference  [1] BACnetObjectPropertyReference,
        issueConfirmedNotifications [2] BOOLEAN,
        timeRemaining               [3] Unsigned,
        covIncrement                [4] REAL OPTIONAL      -- used only with monitored
                                                           -- properties with a datatype of ~~REAL~~numeric
        }
```

[Change **BACnetEventParameters** production in Clause **21**, p.456]

```
BACnetEventParameter ::= CHOICE {

-- These choices have a one-to-one correspondence with the Event_Type enumeration with the exception of the
-- complex-event-type, which is used for proprietary event types.
    change-of-bitstring     [0] SEQUENCE {
                                time-delay              [0] Unsigned,
                                bitmask                 [1] BIT STRING,
                                list-of-bitstring-values [2] SEQUENCE OF BIT STRING
                                },
    …
    unsigned-range          [11] SEQUENCE {
                                time-delay      [0] Unsigned,
                                low-limit       [1] Unsigned,
                                high-limit      [2] Unsigned
                                },
    …
    double-out-of-range     [14] SEQUENCE {
                                time-delay      [0] Unsigned,
                                low-limit       [1] Double,
                                high-limit      [2] Double,
                                deadband        [3] Double
                                },
```

```
    signed-out-of-range         [15] SEQUENCE {
                                    time-delay          [0] Unsigned,
                                    low-limit           [1] INTEGER,
                                    high-limit          [2] INTEGER,
                                    deadband            [3] Unsigned
                                    },
    unsigned-out-of-range       [16] SEQUENCE {
                                    time-delay          [0] Unsigned,
                                    low-limit           [1] Unsigned,
                                    high-limit          [2] Unsigned,
                                    deadband            [3] Unsigned
                                    },
    change-of-characterstring [17] SEQUENCE {
                                    time-delay                  [0] Unsigned,
                                    list-of-alarm-values        [1] SEQUENCE OF CharacterString
                                    }
    }
```

[Change **BACnetEventType** production in Clause **21**, p. 463]

```
BACnetEventType ::= ENUMERATED {
    change-of-bitstring             (0),
    change-of-state                 (1),
    change-of-value                 (2),
    command-failure                 (3),
    floating-limit                  (4),
    out-of-range                    (5),
    -- complex-event-type           (6),  -- see comment below
    -- context tag 7 is deprecated
    change-of-life-safety           (8),
    extended                        (9),
    buffer-ready                    (10),
    unsigned-range                  (11),
    …
    double-out-of-range             (14),
    signed-out-of-range             (15),
    unsigned-out-of-range           (16),
    change-of-characterstring       (17),
    ...
    }
```

[Change **BACnetObjectType** production in Clause **21**, p.457]

```
BACnetObjectType ::= ENUMERATED {
    …
    binary-value                    (5),
    bitstring-value                 (39),
    calendar                        (6),
    command                         (7),
    characterstring-value           (40),
    date-pattern-value              (41),
    date-value                      (42),
    datetime-pattern-value          (43),
    datetime-value                  (44),
    device                          (8),
```

```
…
group                                   (11),
integer-value                           (45),
large-analog-value                      (46),
life-safety-point                       (21),
…
notification-class                      (15),
octetstring-value                       (47),
positive-integer-value                  (48),
program                                 (16),
…
structured-view                         (29),
time-pattern-value                      (49),
time-value                              (50),
trend-log                               (20),
trend-log-multiple                      (27),
…
-- see access-door                      (30),
…
-- see bitstring-value                  (39),
-- see characterstring-value            (40),
-- see date-pattern-value               (41),
-- see date-value                       (42),
-- see datetime-pattern-value           (43),
-- see datetime-value                   (44),
-- see integer-value                    (45),
-- see large-analog-value               (46),
-- see octetstring-value                (47),
-- see positive-Integer-value           (48),
-- see time-pattern-value               (49),
-- see time-value                       (50),

}
```

[Change **BACnetObjectTypesSupported** production in Clause **21**, p.463]

BACnetObjectTypesSupported ::= BIT STRING {

```
    …
    binary-value                (5),
    -- bitstring-value          (39),
    calendar                    (6),
    -- characterstring-value    (40),
    -- credential-data-input    (37),
    command                     (7),
    -- date-pattern-value       (41),
    -- date-value               (42),
    -- datetime-pattern-value   (43),
    -- datetime-value           (44),
    device                      (8),
    …
    group                       (11),
    -- integer-value            (45),
    -- large-analog-value       (46),
    loop                        (12),
    …
    notification-class          (15),
```

```
-- octetstring-value        (47),
-- positive-Integer-value   (48),
program                     (16),
…
-- structured-view          (29),
-- time-pattern-value       (49),
-- time-value               (50),
-- trend-log                (20),
…
bitstring-value             (39),
characterstring-value       (40),
date-pattern-value          (41),
date-value                  (42),
datetime-pattern-value      (43),
datetime-value              (44),
integer-value               (45),
large-analog-value          (46),
octetstring-value           (47),
positive-Integer-value      (48),
time-pattern-value          (49),
time-value                  (50)
}
```

[Change **BACnetNotificationParameters** production in Clause **21**, p.462]

BACnetNotificationParameters ::= CHOICE {
-- These choices have a one-to-one correspondence with the Event_Type enumeration with the exception of the
-- complex-event-type, which is used for proprietary event types.

```
…
out-of-range                [5] SEQUENCE {
                                exceeding-value     [0] REAL,
                                status-flags        [1] BACnetStatusFlags,
                                deadband            [2] REAL,
                                exceeded-limit      [3] REAL
                                },
…
unsigned-range              [11] SEQUENCE {
                                exceeding-value     [0] Unsigned,
                                status-flags        [1] BACnetStatusFlags,
                                exceeded-limit      [2] Unsigned
                                },
…
double-out-of-range         [14] SEQUENCE {
                                exceeding-value     [0] Double,
                                status-flags        [1] BACnetStatusFlags,
                                deadband            [2] Double,
                                exceeded-limit      [2] Double
                                },
signed-out-of-range         [15] SEQUENCE {
                                exceeding-value     [0] INTEGER,
                                status-flags        [1] BACnetStatusFlags,
                                deadband            [2] Unsigned,
                                exceeded-limit      [2] INTEGER
                                },
```

unsigned-out-of-range *[16] SEQUENCE {*
exceeding-value *[0] Unsigned,*
status-flags *[1] BACnetStatusFlags,*
deadband *[2] Unsigned,*
exceeded-limit *[2] Unsigned*
},
change-of-characterstring *[17] SEQUENCE {*
changed-value *[0] CharacterString,*
status-flags *[1] BACnetStatusFlags,*
alarm-value *[2] CharacterString*
}
}

[Add new **BACnetOptionalCharacterString** production in Clause **21**, p.465]

BACnetOptionalCharacterString ::= CHOICE {
null NULL,
characterString CharacterString
}

[Change **BACnetPriorityValue** production in Clause **21**, p.464]

BACnetPriorityValue ::= CHOICE {
null NULL,
real REAL,
~~binary~~*enumerated* ~~BACnetBinaryPV~~*ENUMERATED*,
~~integer~~*unsigned* Unsigned,
boolean *BOOLEAN,*
signed *INTEGER,*
double *Double,*
time *Time,*
characterString *CharacterString,*
octetString *OCTET STRING,*
bitString *BIT STRING,*
date *Date,*
objectid *BACnetObjectIdentifier,*
constructedValue [0] ABSTRACT-SYNTAX.&Type,
datetime *[1] BACnetDateTime*
}

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

BACnetPropertyIdentifier ::= ENUMERATED {
…
bias (14),
bit-mask *(342),*
bit-text *(343),*
buffer-size (126),
…
interval-offset (195),
is-utc ***(344),***
-- issue-confirmed-notifications (51), This property was deleted in version 1 revision 4.
…
-- see secured-status (235),
…

-- see bit-mask (342),
-- see bit-text (343),
-- see is-utc (344)
...
}

[Add new items to **Annex C**, p. 497]

ANNEX C - FORMAL DESCRIPTION OF OBJECT TYPE STRUCTURES (INFORMATIVE)

(This annex is not part of this standard but is included for informative purposes only.)

…

CHARACTERSTRING-VALUE *::= SEQUENCE {*

object-identifier	*[75]*	*BACnetObjectIdentifier,*
object-name	*[77]*	*CharacterString,*
object-type	*[79]*	*BACnetObjectType,*
description	*[28]*	*CharacterString OPTIONAL,*
present-value	*[85]*	*CharacterString,*
status-flags	*[111]*	*BACnetStatusFlags,*
event-state	*[36]*	*BACnetEventState OPTIONAL,*
reliability	*[103]*	*BACnetReliability OPTIONAL,*
out-of-service	*[81]*	*BOOLEAN OPTIONAL,*
time-delay	*[113]*	*Unsigned OPTIONAL,*
priority-array	*[87]*	*BACnetPriorityArray OPTIONAL,*
relinquish-default	*[104]*	*CharacterString OPTIONAL,*
notification-class	*[17]*	*Unsigned OPTIONAL,*
alarm-values	*[7]*	*SEQUENCE OF BACnetOptionalCharacterString OPTIONAL,* *-- accessed as a BACnetARRAY*
fault-value	*[39]*	*SEQUENCE OF BACnetOptionalCharacterString OPTIONAL,* *-- accessed as a BACnetARRAY*
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*

}

DATE-VALUE *::= SEQUENCE {*

object-identifier	*[75]*	*BACnetObjectIdentifier,*
object-name	*[77]*	*CharacterString,*
object-type	*[79]*	*BACnetObjectType,*
description	*[28]*	*CharacterString OPTIONAL,*
present-value	*[85]*	*Date,*
status-flags	*[111]*	*BACnetStatusFlags,*
event-state	*[36]*	*BACnetEventState OPTIONAL,*
reliability	*[103]*	*BACnetReliability OPTIONAL,*
out-of-service	*[81]*	*BOOLEAN OPTIONAL,*
priority-array	*[87]*	*BACnetPriorityArray OPTIONAL,*
relinquish-default	*[104]*	*Date OPTIONAL,*
profile-name	*[168]*	*CharacterString OPTIONAL*

}

DATE-PATTERN-VALUE *::= SEQUENCE {*

object-identifier [75] BACnetObjectIdentifier,
object-name [77] CharacterString,
object-type [79] BACnetObjectType,
description [28] CharacterString OPTIONAL,
present-value [85] Date,
status-flags [111] BACnetStatusFlags,
event-state [36] BACnetEventState OPTIONAL,
reliability [103] BACnetReliability OPTIONAL,
out-of-service [81] BOOLEAN OPTIONAL,
priority-array [87] BACnetPriorityArray OPTIONAL,
relinquish-default [104] Date OPTIONAL,
profile-name [168] CharacterString OPTIONAL
}

DATETIME-VALUE *::= SEQUENCE {*
object-identifier [75] BACnetObjectIdentifier,
object-name [77] CharacterString,
object-type [79] BACnetObjectType,
description [28] CharacterString OPTIONAL,
present-value [85] BACnetDateTime,
status-flags [111] BACnetStatusFlags,
event-state [36] BACnetEventState OPTIONAL,
reliability [103] BACnetReliability OPTIONAL,
out-of-service [81] BOOLEAN OPTIONAL,
priority-array [87] BACnetPriorityArray OPTIONAL,
relinquish-default [104] BACnetDateTime OPTIONAL,
is-utc [344] BOOLEAN,
profile-name [168] CharacterString OPTIONAL
}

DATETIME-PATTERN-VALUE *::= SEQUENCE {*
object-identifier [75] BACnetObjectIdentifier,
object-name [77] CharacterString,
object-type [79] BACnetObjectType,
description [28] CharacterString OPTIONAL,
present-value [85] BACnetDateTime,
status-flags [111] BACnetStatusFlags,
event-state [36] BACnetEventState OPTIONAL,
reliability [103] BACnetReliability OPTIONAL,
out-of-service [81] BOOLEAN OPTIONAL,
priority-array [87] BACnetPriorityArray OPTIONAL,
relinquish-default [104] BACnetDateTime OPTIONAL,
is-utc [344] BOOLEAN,
profile-name [168] CharacterString OPTIONAL
}

LARGE-ANALOG-VALUE *::= SEQUENCE {*
object-identifier [75] BACnetObjectIdentifier,
object-name [77] CharacterString,
object-type [79] BACnetObjectType,
description [28] CharacterString OPTIONAL,
present-value [85] Double,
status-flags [111] BACnetStatusFlags,
event-state [36] BACnetEventState OPTIONAL,
reliability [103] BACnetReliability OPTIONAL,

out-of-service [81] BOOLEAN OPTIONAL,
units [117] BACnetEngineeringUnits,
priority-array [87] BACnetPriorityArray OPTIONAL,
relinquish-default [104] Double OPTIONAL,
cov-increment [22] Double OPTIONAL,
time-delay [113] Unsigned OPTIONAL,
notification-class [17] Unsigned OPTIONAL,
high-limit [45] Double OPTIONAL,
low-limit [59] Double OPTIONAL,
deadband [25] Double OPTIONAL,
limit-enable [52] BACnetLimitEnable 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
}

***BITSTRING-VALUE** ::= SEQUENCE {*
object-identifier [75] BACnetObjectIdentifier,
object-name [77] CharacterString,
object-type [79] BACnetObjectType,
description [28] CharacterString OPTIONAL,
present-value [85] BIT STRING,
bit-mask [342] BIT STRING OPTIONAL,
bit-text [343] SEQUENCE OF CharacterString OPTIONAL,
-- accessed as a BACnetARRAY
status-flags [111] BACnetStatusFlags,
event-state [36] BACnetEventState OPTIONAL,
reliability [103] BACnetReliability OPTIONAL,
out-of-service [81] BOOLEAN OPTIONAL,
priority-array [87] BACnetPriorityArray OPTIONAL,
relinquish-default [104] BIT STRING OPTIONAL,
cov-increment [22] BIT STRING OPTIONAL,
time-delay [113] Unsigned OPTIONAL,
notification-class [17] Unsigned OPTIONAL,
alarm-values [6] SEQUENCE OF BIT STRING OPTIONAL,
-- accessed as a BACnetARRAY
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
}

***OCTETSTRING-VALUE** ::= SEQUENCE {*
object-identifier [75] BACnetObjectIdentifier,
object-name [77] CharacterString,
object-type [79] BACnetObjectType,
description [28] CharacterString OPTIONAL,
present-value [85] OCTET STRING,
status-flags [111] BACnetStatusFlags,
event-state [36] BACnetEventState OPTIONAL,
reliability [103] BACnetReliability OPTIONAL,

out-of-service *[81]* *BOOLEAN OPTIONAL,*
priority-array *[87]* *BACnetPriorityArray OPTIONAL,*
relinquish-default *[104]* *OCTET STRING OPTIONAL,*
profile-name *[168]* *CharacterString OPTIONAL*
}

TIME-VALUE *::= SEQUENCE {*
object-identifier *[75]* *BACnetObjectIdentifier,*
object-name *[77]* *CharacterString,*
object-type *[79]* *BACnetObjectType,*
description *[28]* *CharacterString OPTIONAL,*
present-value *[85]* *Time,*
status-flags *[111]* *BACnetStatusFlags,*
event-state *[36]* *BACnetEventState OPTIONAL,*
reliability *[103]* *BACnetReliability OPTIONAL,*
out-of-service *[81]* *BOOLEAN OPTIONAL,*
priority-array *[87]* *BACnetPriorityArray OPTIONAL,*
relinquish-default *[104]* *Time OPTIONAL,*
profile-name *[168]* *CharacterString OPTIONAL*
}

TIME-PATTERN-VALUE *::= SEQUENCE {*
object-identifier *[75]* *BACnetObjectIdentifier,*
object-name *[77]* *CharacterString,*
object-type *[79]* *BACnetObjectType,*
description *[28]* *CharacterString OPTIONAL,*
present-value *[85]* *Time,*
status-flags *[111]* *BACnetStatusFlags,*
event-state *[36]* *BACnetEventState OPTIONAL,*
reliability *[103]* *BACnetReliability OPTIONAL,*
out-of-service *[81]* *BOOLEAN OPTIONAL,*
priority-array *[87]* *BACnetPriorityArray OPTIONAL,*
relinquish-default *[104]* *Time OPTIONAL,*
profile-name *[168]* *CharacterString OPTIONAL*
}

INTEGER-VALUE *::= SEQUENCE {*
object-identifier *[75]* *BACnetObjectIdentifier,*
object-name *[77]* *CharacterString,*
object-type *[79]* *BACnetObjectType,*
description *[28]* *CharacterString OPTIONAL,*
present-value *[85]* *INTEGER,*
status-flags *[111]* *BACnetStatusFlags,*
event-state *[36]* *BACnetEventState OPTIONAL,*
reliability *[103]* *BACnetReliability OPTIONAL,*
out-of-service *[81]* *BOOLEAN OPTIONAL,*
units *[117]* *BACnetEngineeringUnits,*
priority-array *[87]* *BACnetPriorityArray OPTIONAL,*
relinquish-default *[104]* *INTEGER OPTIONAL,*
cov-increment *[22]* *Unsigned OPTIONAL,*
time-delay *[113]* *Unsigned OPTIONAL,*
notification-class *[17]* *Unsigned OPTIONAL,*
high-limit *[45]* *INTEGER OPTIONAL,*
low-limit *[59]* *INTEGER OPTIONAL,*
deadband *[25]* *Unsigned OPTIONAL,*
limit-enable *[52]* *BACnetLimitEnable 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
}

POSITIVE-INTEGER-VALUE *::= SEQUENCE {*
object-identifier [75] BACnetObjectIdentifier,
object-name [77] CharacterString,
object-type [79] BACnetObjectType,
description [28] CharacterString OPTIONAL,
present-value [85] Unsigned,
status-flags [111] BACnetStatusFlags,
event-state [36] BACnetEventState OPTIONAL,
reliability [103] BACnetReliability OPTIONAL,
out-of-service [81] BOOLEAN OPTIONAL,
units [117] BACnetEngineeringUnits,
priority-array [87] BACnetPriorityArray OPTIONAL,
relinquish-default [104] Unsigned OPTIONAL,
cov-increment [22] Unsigned OPTIONAL,
time-delay [113] Unsigned OPTIONAL,
notification-class [17] Unsigned OPTIONAL,
high-limit [45] Unsigned OPTIONAL,
low-limit [59] Unsigned OPTIONAL,
deadband [25] Unsigned OPTIONAL,
limit-enable [52] BACnetLimitEnable 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 new items to **Annex D**, p. 512]

D.A Example of a CharacterString Value object

Property: Object_Identifier = (CharacterString_Value, Instance 1)
Property: Object_Name = "SOMEIMPORTANTVALUE"
Property: Object_Type = CHARACTERSTRING_VALUE
Property: Description = "Some Description"
Property: Present_Value = "Some String Value"
Property: Status_Flags = {FALSE, FALSE, FALSE, FALSE}
Property: Event_State = NORMAL
Property: Reliability = NO_FAULT_DETECTED
Property: Out_Of_Service = FALSE
Property: Priority_Array= {NULL, NULL, …, "Some String Value"}
Property: Relinquish_Default= ""
Property: Time_Delay = 10
Property: Notification_Class = 3
Property: Alarm_Values = {"ALM","FLT"}
Property: Event_Enable = {TRUE,FALSE,TRUE}
Property: Acked_Transitions = {TRUE,TRUE,TRUE}

Property:	Notify_Type =	ALARM
Property:	Event_Time_Stamps =	((23-MAR-05,18:50:21.2), (*-*-*,*:*:*.*), (23-MAR-05,19:01:34.0))

D.B Example of a DateTime Value object

Property:	Object_Identifier =	(DateTime_Value, Instance 1)
Property:	Object_Name =	"SOMEIMPORTANTVALUE"
Property:	Object_Type =	DATETIME_VALUE
Property:	Description =	"Some Description"
Property:	Present_Value =	(23-MAR-1998,12:32:33.0),
Property:	Status_Flags =	{FALSE, FALSE, FALSE, FALSE}
Property:	Event_State =	NORMAL
Property:	Reliability =	NO_FAULT_DETECTED
Property:	Is_UTC =	FALSE
Property:	Out_Of_Service =	FALSE

D.C Example of a Large Analog Value object

Property:	Object_Identifier =	(Large_Analog_Value, Instance 1)
Property:	Object_Name =	"SOMEIMPORTANTVALUE"
Property:	Object_Type =	LARGE_ANALOG_VALUE
Property:	Description =	"Some Description"
Property:	Present_Value =	123456.789123456
Property:	Status_Flags =	{FALSE, FALSE, FALSE, FALSE}
Property:	Event_State =	NORMAL
Property:	Reliability =	NO_FAULT_DETECTED
Property:	Out_Of_Service =	FALSE
Property:	Priority_Array=	{NULL ,NULL, …, 123456.789123456}
Property:	Relinquish_Default=	50.0
Property:	Time_Delay =	10
Property:	Notification_Class =	3
Property:	High_Limit =	60.0
Property:	Low_Limit =	55.0
Property:	Deadband =	1.0
Property:	Limit_Enable =	{TRUE,TRUE}
Property:	Event_Enable =	{TRUE,FALSE,TRUE}
Property:	Acked_Transitions =	{TRUE,TRUE,TRUE}
Property:	Notify_Type =	ALARM
Property:	Event_Time_Stamps =	((23-MAR-05,18:50:21.2), (*-*-*,*:*:*.*), (23-MAR-05,19:01:34.0))

D.D Example of a BitString Value object

Property:	Object_Identifier =	(BitString_Value, Instance 1)
Property:	Object_Name =	"SOMEIMPORTANTVALUE"
Property:	Object_Type =	BITSTRING_VALUE
Property:	Description =	"Some Description"
Property:	Present_Value =	{FALSE,TRUE,FALSE}
Property:	Bit_Text =	{"Overheated","Needs Oil","Change Filter"}
Property:	Status_Flags =	{FALSE, FALSE, FALSE, FALSE}
Property:	Event_State =	NORMAL
Property:	Reliability =	NO_FAULT_DETECTED

Property:	Out_Of_Service =	FALSE
Property:	COV_Increment =	{TRUE,TRUE,FALSE}
Property:	Time_Delay =	10
Property:	Notification_Class =	3
Property:	Alarm_Values =	{{TRUE,TRUE,TRUE}}
Property:	Bit_Mask =	{TRUE,TRUE,TRUE}
Property:	Event_Enable =	{TRUE,FALSE,TRUE}
Property:	Acked_Transitions =	{TRUE,TRUE,TRUE}
Property:	Notify_Type =	ALARM
Property:	Event_Time_Stamps =	((23-MAR-05,18:50:21.2), (*-*-*,*:*:*.*), (23-MAR-05,19:01:34.0))

D.E Example of an OctetString Value object

Property:	Object_Identifier =	(OctetString_Value, Instance 1)
Property:	Object_Name =	"SOMEIMPORTANTVALUE"
Property:	Object_Type =	OCTETSTRING_VALUE
Property:	Description =	"Some Description"
Property:	Present_Value =	{1,27,49,5,137}
Property:	Status_Flags =	{FALSE, FALSE, FALSE, FALSE}
Property:	Event_State =	NORMAL
Property:	Reliability =	NO_FAULT_DETECTED
Property:	Out_Of_Service =	FALSE

D.F Example of a Time Value object

Property:	Object_Identifier =	(Time_Value, Instance 1)
Property:	Object_Name =	"SOMEIMPORTANTVALUE"
Property:	Object_Type =	TIME_VALUE
Property:	Description =	"Some Description"
Property:	Present_Value =	12:34:56.77
Property:	Status_Flags =	{FALSE, FALSE, FALSE, FALSE}
Property:	Event_State =	NORMAL
Property:	Reliability =	NO_FAULT_DETECTED
Property:	Out_Of_Service =	FALSE
Property:	Priority_Array=	{NULL ,NULL, …, 12:34:56.77}
Property:	Relinquish_Default=	00:00:00.00

D.G Example of a Integer Value object

Property:	Object_Identifier =	(Integer_Value, Instance 1)
Property:	Object_Name =	"SOMEIMPORTANTVALUE"
Property:	Object_Type =	INTEGER_VALUE
Property:	Description =	"Some Description"
Property:	Present_Value =	-1238
Property:	Status_Flags =	{FALSE, FALSE, FALSE, FALSE}
Property:	Event_State =	NORMAL
Property:	Reliability =	NO_FAULT_DETECTED
Property:	Out_Of_Service =	FALSE
Property:	Priority_Array=	{NULL ,NULL, ..., 27... NULL}
Property:	Relinquish_Default=	-52
Property:	Time_Delay =	10
Property:	Notification_Class =	3
Property:	High_Limit =	600

Property:	Low_Limit =	50
Property:	Deadband =	10
Property:	Limit_Enable =	{TRUE,TRUE}
Property:	Event_Enable =	{TRUE,FALSE,TRUE}
Property:	Acked_Transitions =	{TRUE,TRUE,TRUE}
Property:	Notify_Type =	ALARM
Property:	Event_Time_Stamps =	((23-MAR-05,18:50:21.2), (*-*-*,*:*:*.*), (23-MAR-05,19:01:34.0))

D.H Example of an Positive Integer Value object

Property:	Object_Identifier =	(Positive Integer_Value, Instance 1)
Property:	Object_Name =	"SOMEIMPORTANTVALUE"
Property:	Object_Type =	POSITIVE_INTEGER_VALUE
Property:	Description =	"Some Description"
Property:	Present_Value =	123456789
Property:	Status_Flags =	{FALSE, FALSE, FALSE, FALSE}
Property:	Event_State =	NORMAL
Property:	Reliability =	NO_FAULT_DETECTED
Property:	Out_Of_Service =	FALSE
Property:	Priority_Array=	{NULL ,NULL, …, 123456789}
Property:	Relinquish_Default=	0
Property:	Time_Delay =	10
Property:	Notification_Class =	3
Property:	High_Limit =	60000000
Property:	Low_Limit =	0
Property:	Deadband =	100
Property:	Limit_Enable =	{TRUE,FALSE}
Property:	Event_Enable =	{TRUE,FALSE,TRUE}
Property:	Acked_Transitions =	{TRUE,TRUE,TRUE}
Property:	Notify_Type =	ALARM
Property:	Event_Time_Stamps =	((23-MAR-05,18:50:21.2), (*-*-*,*:*:*.*), (23-MAR-05,19:01:34.0))

D.I Example of a Date Value object

Property:	Object_Identifier =	(Date_Value, Instance 1)
Property:	Object_Name =	"SOMEIMPORTANTVALUE"
Property:	Object_Type =	DATE_VALUE
Property:	Description =	"Some Description"
Property:	Present_Value =	(23-MAR-1998),
Property:	Status_Flags =	{FALSE, FALSE, FALSE, FALSE}
Property:	Event_State =	NORMAL
Property:	Reliability =	NO_FAULT_DETECTED
Property:	Out_Of_Service =	FALSE
Property:	Priority_Array=	{NULL ,NULL, …, (23-MAR-1998)}
Property:	Relinquish_Default=	(*-*-*)

D.J Example of a DateTime Pattern Value object

Property:	Object_Identifier =	(DateTime_Pattern_Value, Instance 1)
Property:	Object_Name =	"SOMEIMPORTANTVALUE"
Property:	Object_Type =	DATETIME_PATTERN_VALUE

Property:	Description =	"Some Description"
Property:	Present_Value =	(23-MAR-*,12:*:*.*),
Property:	Status_Flags =	{FALSE, FALSE, FALSE, FALSE}
Property:	Event_State =	NORMAL
Property:	Reliability =	NO_FAULT_DETECTED
Property:	Out_Of_Service =	FALSE

D.K Example of a Time Pattern Value object

Property:	Object_Identifier =	(Time_Pattern_Value, Instance 1)
Property:	Object_Name =	"SOMEIMPORTANTVALUE"
Property:	Object_Type =	TIME_PATTERN_VALUE
Property:	Description =	"Some Description"
Property:	Present_Value =	12:*:*.*
Property:	Status_Flags =	{FALSE, FALSE, FALSE, FALSE}
Property:	Event_State =	NORMAL
Property:	Reliability =	NO_FAULT_DETECTED
Property:	Out_Of_Service =	FALSE
Property:	Priority_Array=	{NULL ,NULL, …, 12:*:*.*}
Property:	Relinquish_Default=	00:00:00.00

D.L Example of a Date Pattern Value object

Property:	Object_Identifier =	(Date_Pattern_Value, Instance 1)
Property:	Object_Name =	"SOMEIMPORTANTVALUE"
Property:	Object_Type =	DATE_PATTERN_VALUE
Property:	Description =	"Some Description"
Property:	Present_Value =	(23-MAR-*),
Property:	Status_Flags =	{FALSE, FALSE, FALSE, FALSE}
Property:	Event_State =	NORMAL
Property:	Reliability =	NO_FAULT_DETECTED
Property:	Out_Of_Service =	FALSE
Property:	Priority_Array=	{NULL ,NULL, …, (23-MAR-*)}
Property:	Relinquish_Default=	(*-*-*)

[Add a new entry to **History of Revisions**, p. 688]

(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

Protocol		***Summary of Changes to the Standard***
Version	***Revision***	
…	…	**…**
1	10	**Addendum *w* to ANSI/ASHRAE 135-2008** Approved by the ASHRAE Standards Committee January 23, 2010; by the ASHRAE Board of Directors January 27, 2010; and by the American National Standards Institute January 28, 2010. 1. Add more primitive value objects.

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.