12.24 Schedule Object Type

The Schedule object type defines a standardized object used to describe a periodic schedule that may recur during a range of dates, with optional exceptions at arbitrary times on arbitrary dates. The Schedule object also serves as a binding between these scheduled times and the writing of specified "values" to specific properties of specific objects at those times. The Schedule object type and its properties are summarized in Table 12-28 and described in detail in this subclause.

Schedules are divided into days, of which there are two types: normal days within a week and exception days. Both types of days can specify scheduling events for either the full day or portions of a day, and a priority mechanism defines which scheduled event is in control at any given time.

The current state of the Schedule object is represented by the value of its Present_Value property, which is normally calculated using the time/value pairs from the Weekly_Schedule and Exception_Schedule properties, with a default value for use when no schedules are in effect. Details of this calculation are provided in the description of the Present_Value property.

Versions of the Schedule object prior to Protocol_Revision 4 only support schedules that define an entire day, from midnight to midnight. For compatibility with these versions, this whole day behavior can be achieved by using a specific schedule format. Weekly_Schedule and Exception_Schedule values that begin at 00:00, and do not use any NULL values, will define schedules for the entire day. Property values in this format will produce the same results in all versions of the Schedule object.

Table 12-28. Properties of the Schedule Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Present_Value	Any	R
Description	CharacterString	O
Effective_Period	BACnetDateRange	R
Weekly_Schedule	BACnetARRAY[7]of BACnetDailySchedule	O[1]
Exception_Schedule	BACnetARRAY[N]of BACnetSpecialEvent	O[1]
Schedule_Default	Any	R
List_Of_Object_Property_References	List of BACnetDeviceObjectPropertyReference	R
Priority_For_Writing	Unsigned(1..16)	R
Status_Flags	BACnetStatusFlags	R
Reliability	BACnetReliability	R
Out_Of_Service	BOOLEAN	R
Profile_Name	CharacterString	O

[1] At least one of these properties is required**.**

12.24.1 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.

12.24.2 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.

12.24.3 Object_Type

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

12.24.4 Present_Value

This property indicates the current value of the schedule, which may be any primitive datatype. As a result, most analog, binary, and enumerated values may be scheduled. This property shall be writable when Out_Of_Service is TRUE (see 12.24.14).

Any change in the value of this property shall be written to all members of the List_Of_Object_Property_References property. An error writing to any member of the list shall not stop the Schedule object from writing to the remaining members.

The normal calculation of the value of the Present_Value property is illustrated as follows (the actual algorithm used is a local matter but must yield the same results as this one):

1. Find the highest relative priority (as defined by 12.24.8) Exception_Schedule array element that is in effect for the current day and whose current value (see method below) is not NULL, and assign that value to the Present_Value property.
2. If the Present_Value was not assigned in the previous step, then evaluate the current value of the Weekly_Schedule array element for the current day and if that value is not NULL, assign it to the Present_Value property.
3. If the Present_Value was not assigned in the previous steps, then assign the value of the Schedule_Default property to the Present_Value property.

The method for evaluating the current value of a schedule (either exception or weekly) is to find the latest element in the list of BACnetTimeValues that occurs on or before the current time, and then use that element's value as the current value for the schedule. If no such element is found, then the current value for the schedule shall be NULL.

These calculations are such that they can be performed at any time and the correct value of Present_Value property will result. These calculations must be performed at 00:00 each day, whenever the device resets, whenever properties that can affect the results are changed, whenever the time in the device changes by an amount that may have an effect on the calculation result, and at other times, as required, to maintain the correct value of the Present_Value property through the normal passage of time.

Note that the Present_Value property will be assigned the value of the Schedule_Default property at 00:00 of any given day, unless there is an entry for 00:00 in effect for that day. If a scheduled event logically begins on one day and ends on another, an entry at 00:00 shall be placed in the schedule that is in effect for the second day, and for any subsequent days of the event's duration, to ensure the correct result whenever Present_Value is calculated

12.24.5 Description

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

12.24.6 Effective_Period

This property specifies the range of dates within which the Schedule object is active. Seasonal scheduling may be achieved by defining several SCHEDULE objects with non-overlapping Effective_Periods to control the same property references. Upon entering its effective period, the object shall calculate its Present_Value and write that value to all members of the List_Of_Object_Property_References property. An error writing to any member of the list shall not stop the Schedule object from writing to the remaining members.

12.24.7 Weekly_Schedule

This property is a BACnetARRAY containing exactly seven elements. Each of the elements 1-7 contains a BACnetDailySchedule. A BACnetDailySchedule consists of a list of BACnetTimeValues that are (time, value) pairs, which describe the sequence of schedule actions on one day of the week when no Exception_Schedule is in effect. The array elements 1-7 correspond to the days Monday - Sunday, respectively. The Weekly_Schedule is an optional property, but either the Weekly_Schedule or a non-empty Exception_Schedule shall be supported in every instance of a Schedule object.

If the Weekly_Schedule property is written with a schedule item containing a datatype not supported by this instance of the Schedule object (e.g., the List_Of_Object_Property_References property cannot be configured to reference a property of the unsupported datatype), the device may return a Result(-) response, specifying an 'Error Class' of PROPERTY and an 'Error Code' of DATATYPE_NOT_SUPPORTED.

12.24.8 Exception_Schedule

This property is a BACnetARRAY of BACnetSpecialEvents. Each BACnetSpecialEvent describes a sequence of schedule actions that takes precedence over the normal day's behavior on a specific day or days.

BACnetSpecialEvent ::= (Period, List of BACnetTimeValue, EventPriority)

Period ::= Choice of {BACnetCalendarEntry | CalendarReference}

EventPriority ::= Unsigned (1..16)

The Period may be a BACnetCalendarEntry or it may refer to a Calendar object. A BACnetCalendarEntry would be used if the Exception_Schedule is specific to this Schedule object, while calendars might be defined for common holidays to be referenced by multiple Schedule objects. Each BACnetCalendarEntry is either an individual date (Date), range of dates (BACnetDateRange), or month/week-of-month/day-of-week specification (BACnetWeekNDay). If the current date matches any of the calendar entry criteria, the Exception Schedule would be activated and the list of BACnetTimeValues would be enabled for use.

Individual fields of the various constructs of the BACnetCalendarEntry may also have a "wildcard" value used for determining if the current date falls within the Period of the Exception Schedule. In a date range, for example, if the startDate is a wildcard, it means "any date up to and including the endDate." If the endDate is a wildcard, it means "any date from the startDate on." If the calendar entry were a BACnetWeekNDay with wildcard for month and week-of-month fields but with a specific day-of-week, it would mean the Exception Schedule would apply on that day-of-week all year long.

Each BACnetSpecialEvent contains an EventPriority that determines its importance relative to other BACnetSpecialEvents within the same Exception_Schedule. Since SpecialEvents within the same Exception_Schedule may have overlapping periods, it is necessary to have a mechanism to determine the relative priorities for the SpecialEvents that apply on any given day. If more than one SpecialEvent applies to a given day, the relative priority of the SpecialEvents shall be determined by their EventPriority values. If multiple overlapping SpecialEvents have the same EventPriority value, then the SpecialEvent with the lowest index number in the array shall have higher relative priority. The highest EventPriority is 1 and the lowest is 16. The EventPriority is not related to the Priority_For_Writing property of the Schedule object.

If a BACnet Device supports writing to the Exception_Schedule property, all possible choices in the BACnetSpecialEvents shall be supported. If the size of this array is increased by writing to array index zero, each new array element shall contain an empty List of BACnetTimeValue.

If the Exception_Schedule property is written with a schedule item containing a datatype not supported by this instance of the Schedule object (e.g., the List_Of_Object_Property_References property cannot be configured to reference a property of the unsupported datatype), the device may return a Result(-) response, specifying an 'Error Class' of PROPERTY and an 'Error Code' of DATATYPE_NOT_SUPPORTED.

12.24.9 Schedule_Default

This property holds a default value to be used for the Present_Value property when no other scheduled value is in effect (see 12.24.4). This may be any primitive datatype.

If the Schedule_Default property is written with a value containing a datatype not supported by this instance of the Schedule object (e.g., the List_Of_Object_Property_References property cannot be configured to reference a property of the unsupported datatype), the device may return a Result(-) response, specifying an 'Error Class' of PROPERTY and an 'Error Code' of DATATYPE_NOT_SUPPORTED.

12.24.10 List_Of_Object_Property_Reference

This property specifies the Device Identifiers, Object Identifiers and Property Identifiers of the properties to be written with specific values at specific times on specific days.

If this property is writable, it may be restricted to only support references to objects inside of the device containing the Schedule object. If the property is restricted to referencing objects within the containing device, an attempt to write a reference to an object outside the containing device into this property shall cause a Result(-) to be returned with an error class of PROPERTY and an error code of OPTIONAL_FUNCTIONALITY_NOT_SUPPORTED.

If this property is set to reference an object outside the device containing the Schedule object, the method used for writing to the referenced property value for the purpose of controlling the property is a local matter. The only restriction on the method of writing to the referenced property is that the scheduling device be capable of using WriteProperty for this purpose so as to be interoperable with all BACnet devices.

12.24.11 Priority_For_Writing

This property defines the priority at which the referenced properties are commanded. It corresponds to the 'Priority' parameter of the WriteProperty service. It is an unsigned integer in the range 1-16, with 1 being considered the highest priority and 16 the lowest. See Clause 19.

12.24.12 Status_Flags

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

{IN_ALARM, FAULT, OVERRIDDEN, OUT_OF_SERVICE}

where:

IN_ALARM The value of this flag shall be logical FALSE (0).

FAULT Logical TRUE (1) if the Reliability property does not have a value of NO_FAULT_DETECTED, otherwise logical FALSE (0).

OVERRIDDEN Logical TRUE (1) if the schedule object 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 has a value of TRUE, otherwise logical FALSE (0).

12.24.13 Reliability

The Reliability property, of type BACnetReliability, provides an indication that the properties of the schedule object are in a consistent state. All non-NULL values used in the Weekly_Schedule, the Exception_Schedule, and the Schedule_Default properties shall be of the same datatype, and all members of the List_Of_Object_Property_References shall be writable with that datatype. If these conditions are not met, then this property shall have the value CONFIGURATION_ERROR. The Reliability property for this object type may have any of the following values:

{NO_FAULT_DETECTED, CONFIGURATION_ERROR, UNRELIABLE_OTHER}.

If the List_Of_Object_Property_References contains a member that references a property in a remote device, the detection of a configuration error may be delayed until an attempt is made to write a scheduled value.

12.24.14 Out_Of_Service

The Out_Of_Service property, of type BOOLEAN, is an indication whether (TRUE) or not (FALSE) the internal calculations of the schedule object are used to determine the value of the Present_Value property. This means that the Present_Value property is decoupled from the internal calculations and will not track changes to other properties when Out_Of_Service is TRUE. Other functions that depend on the state of the Present_Value, such as writing to the members of the List_Of_Object_Property_References, shall respond to changes made to that property while Out_Of_Service is TRUE, as if those changes had occurred by internal calculations.

12.24.15 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.

12.25 Trend Log Object Type

A Trend Log object monitors a property of a referenced object and, when predefined conditions are met, saves ("logs") the value of the property and a timestamp in an internal buffer for subsequent retrieval. The data may be logged periodically or upon a change of value. Errors that prevent the acquisition of the data, as well as changes in the status or operation of the logging process itself, are also recorded. Each timestamped buffer entry is called a trend log "record."

The referenced object may reside in the same device as the Trend Log object or in an external device. The referenced property's value may be recorded upon COV subscription or periodic poll. Where status flags are available (such as when the COVNotification or ReadPropertyMultiple services are used), they are also acquired and saved with the data.

Each Trend Log object maintains an internal, optionally fixed-size buffer. This buffer fills or grows as log records are added. If the buffer becomes full, the least recent record is overwritten when a new record is added, or collection may be set to stop. Trend Log records are transferred as BACnetLogRecords using the ReadRange service. The buffer may be cleared by writing a zero to the Record_Count property. Each record in the buffer has an implied SequenceNumber which is equal to the value the Total_Record_Count property has immediately after the record is added. If the Total_Record_Count is incremented past 2^{32}-1, then it shall reset to 1.

Several datatypes are defined for storage in the log records. The ability to store ANY datatypes is optional. Data stored in the log buffer may be optionally restricted in size to 32 bits, as in the case of bit strings, to facilitate implementation in devices with strict storage requirements.

Logging may be enabled and disabled through the Log_Enable property and at dates and times specified by the Start_Time and Stop_Time properties. Trend Log enabling and disabling is recorded in the log buffer.

Event reporting (notification) may be provided to facilitate automatic fetching of log records by processes on other devices such as fileservers. Support is provided for algorithmic reporting; optionally, intrinsic reporting may be provided.

In intrinsic reporting, when the number of records specified by the Notification_Threshold property have been collected since the previous notification (or startup), a new notification is sent to all subscribed devices. BUFFER_READY algorithmic reporting is described in Clause 13.3.7.

In response to a notification, subscribers may fetch all of the new records. If a subscriber needs to fetch all of the new records, it should use the 'By Sequence Number' form of the ReadRange service request.

A missed notification may be detected by a subscriber if the Current_Notify_Record it received in its previous notification is different than the Previous_Notify_Record parameter of the current notification. If the ReadRange-ACK response to the ReadRange request issued under these conditions has its 'first-item' flag set to TRUE, Trend Log records have probably been missed by this subscriber.

The acquisition of log records by remote devices has no effect upon the state of the Trend Log object itself. This allows completely independent, but properly sequential, access to its log records by all remote devices. Any remote device can independently update its records at any time.

Table 12-29. Properties of the Trend Log Object Type

Property Identifier	Property Datatype	Conformance Code
Object_Identifier	BACnetObjectIdentifier	R
Object_Name	CharacterString	R
Object_Type	BACnetObjectType	R
Description	CharacterString	O
Log_Enable	BOOLEAN	W
Start_Time	BACnetDateTime	O[1,2]
Stop_Time	BACnetDateTime	O[1,2]
Log_DeviceObjectProperty	BACnetDeviceObjectPropertyReference	O[1]
Log_Interval	Unsigned	O[1,2]
COV_Resubscription_Interval	Unsigned	O
Client_COV_Increment	BACnetClientCOV	O
Stop_When_Full	BOOLEAN	R
Buffer_Size	Unsigned32	R
Log_Buffer	List of BACnetLogRecord	R
Record_Count	Unsigned32	W
Total_Record_Count	Unsigned32	R
Notification_Threshold	Unsigned32	O[3]
Records_Since_Notification	Unsigned32	O[3]
Last_Notify_Record	Unsigned32	O[3]
Event_State	BACnetEventState	R
Notification_Class	Unsigned	O[3]
Event_Enable	BACnetEventTransitionBits	O[3]
Acked_Transitions	BACnetEventTransitionBits	O[3]
Notify_Type	BACnetNotifyType	O[3]
Event_Time_Stamps	BACnetARRAY[3] of BACnetTimeStamp	O[3]
Profile_Name	CharacterString	O

[1] These properties are required to be present if the monitored property is a BACnet property.
[2] If present, these properties are required to be writable.
[3] These properties are required to be present if the object supports intrinsic reporting.

12.25.1 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.

12.25.2 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.

12.25.3 Object_Type

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

12.25.4 Description

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

12.25.5 Log_Enable

This property, of type BOOLEAN, indicates and controls whether (TRUE) or not (FALSE) logging is enabled. A value of FALSE overrides the time interval defined by Start_Time and Stop_Time. If logging is otherwise enabled by the Start_Time and Stop_Time properties, changes to the value of the Log_Enable property shall be recorded in the log. When the device begins operation the value TRUE shall be recorded in the log.

12.25.6 Start_Time

This property, of type BACnetDateTime, specifies the date and time at or after which logging shall be enabled by this property. If any of the fields of the BACnetDateTime contain "wildcard" values, then the conditions for logging to be enabled by Start_Time shall be ignored. If Start_Time specifies a date and time after Stop_Time, then logging shall be disabled. This property must be writable if present.

12.25.7 Stop_Time

This property, of type BACnetDateTime, specifies the date and time at or after which logging shall be disabled by this property. If any of the fields of the BACnetDateTime contain "wildcard" values, then the conditions for logging to be enabled by Stop_Time shall be ignored. If Stop_Time specifies a date and time earlier than Start_Time, then logging shall be disabled. This property must be writable if present.

12.25.8 Log_DeviceObjectProperty

This property, of type BACnetDeviceObjectPropertyReference, specifies the Device Identifier, Object Identifier and Property Identifier of the property to be trend logged.

If this property is writable, it may be restricted to reference only objects inside the device containing the Trend Log object. If the property is restricted to referencing objects within the containing device, an attempt to write a reference to an object outside the containing device into this property shall cause a Result(-) to be returned.

12.25.9 Log_Interval

This property, of type Unsigned, specifies the periodic interval in hundredths of seconds for which the referenced property is to be logged. If this property has the value zero then the Trend Log shall issue COV subscriptions for the referenced property. The value of this property must be non-zero if COV_Resubscription_Interval is not present. This property must be writable if present.

12.25.10 COV_Resubscription_Interval

If the Trend Log is acquiring data from a remote device by COV subscription, this property, of type Unsigned, specifies the number of seconds between COV resubscriptions, provided that COV subscription is in effect. SubscribeCOV requests shall specify twice this lifetime for the subscription and shall specify the issuance of confirmed notifications. If COV subscriptions are in effect, the first COV subscription is issued when the Trend Log object begins operation or when Log_Enable becomes TRUE. If present, the value of this property must be non-zero. If this property is not present, then COV subscription shall not be attempted.

12.25.11 Client_COV_Increment

If the Trend Log is acquiring COV data, this property, of type BACnetClientCOV, specifies the increment to be used in determining that a change of value has occurred. If the referenced object and property supports COV reporting according to 13.1, this property may have the value NULL; in this case change of value is determined by the criteria of 13.1.

12.25.12 Stop_When_Full

This property, of type BOOLEAN, specifies whether (TRUE) or not (FALSE) logging should cease when the buffer is full. When logging ceases, Log_Enable shall be set FALSE.

12.25.13 Buffer_Size

This property, of type Unsigned32, shall specify the maximum number of records the buffer may hold. If writable, it may not be written when Log_Enable is TRUE. The disposition of existing records when Buffer_Size is written is a local matter.

12.25.14 Log_Buffer

This property is a list of up to Buffer_Size timestamped records of datatype BACnetLogRecord, each of which conveys a recorded data value, an error related to data-collection, or status changes in the Trend Log object. Each record has data fields as follows:

Timestamp The local date and time when the record was collected.

LogDatum The data value read from the monitored object and property, an error encountered in an attempt to read a value, or a change in status or operation of the Trend Log object itself.

StatusFlags The Status_Flags property of the monitored object, if present and available atomically associated with the LogDatum data value. If the Status_Flags property is not present or not available atomically associated with the data value, this item shall not be included in the log record.

The choices available for the LogDatum are listed below:

log-status This choice represents a change in the status or operation of the Trend Log object. Whenever one of the events represented by the flags listed below occurs, except as noted, a record shall be appended to the buffer.

log-disabled This flag is set whenever the Trend Log object is disabled, such as when Log_Enable is set to FALSE. Whenever the Trend Log object begins operation, this flag shall be presumed to have changed from TRUE to FALSE and a log entry shall be made.

buffer-purged This flag shall be set to TRUE whenever the buffer is deleted by a write of the value zero to the Record_Count property. After this value is recorded in the buffer, the subsequent immediate change to FALSE shall not be recorded.

boolean-value
real-value
enum-value
unsigned-value
signed-value
bitstring-value
null-value These choices represent data values read from the monitored object and property.

failure This choice represents an error encountered in an attempt to read a data value from the monitored object. If the error is conveyed by an error response from a remote device the Error Class and Error Code in the response shall be recorded.

time-change This choice represents a change in the clock setting in the device, it records the number of seconds by which the clock changed. If the number is not known, such as when the clock is initialized for the first time, the value recorded shall be zero.

any-value This choice represents data values read from the monitored object and property.

Also associated with each record is an implied record number, the value of which is equal to Total_Record_Count at the point where the record has been added into the Log Buffer and Total_Record_Count has been adjusted accordingly. All clients must be able to correctly handle the case where the Trend Log is reset such that its Total_Record_Count is returned to zero and also the case where Total_Record_Count has wrapped back to 1.

The buffer is not network accessible except through the use of the ReadRange service, in order to avoid problems with record sequencing when segmentation is required.

12.25.15 Record_Count

This property, of type Unsigned32, shall represent the number of records currently resident in the log buffer. A write of the value zero to this property shall cause all records in the log buffer to be deleted and Records_Since_Notification to be reset to zero. Upon completion, this event shall be reported in the log as the initial entry.

12.25.16 Total_Record_Count

This property, of type Unsigned32, shall represent the total number of records collected by the Trend Log object since creation. When the value of Total_Record_Count reaches its maximum possible value of $2^{32} – 1$, the next value it takes shall be one. Once this value has wrapped to one, its semantic value (the total number of records collected) has been lost but its use in generating notifications remains.

12.25.17 Notification_Threshold

This property, of type Unsigned32, shall specify the value of Records_Since_Notification at which notification occurs. This property is required if intrinsic reporting is supported by this object.

12.25.18 Records_Since_Notification

This property, of type Unsigned32, represents the number of records collected since the previous notification, or since the beginning of logging if no previous notification has occurred. This property is required if intrinsic reporting is supported by this object.

12.25.19 Last_Notify_Record

This property, of type Unsigned32, represents the SequenceNumber associated with the most recently collected record whose collection triggered a notification. If no notification has occurred since logging began the value of this property shall be zero. This property is required if intrinsic reporting is supported by this object.

12.25.20 Event_State

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

12.25.21 Notification_Class

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

12.25.22 Event_Enable

This property, of type BACnetEventTransitionBits, shall convey three flags that separately enable and disable reporting of TO-FAULT and TO-NORMAL events. In the context of Trend Log objects, the value of the Records_Since_Notification property becoming equal to or greater than the value of the Notification_Threshold property shall cause a TO-NORMAL transition. The failure of an attempted COV subscription shall cause a TO-FAULT state transition. The TO-NORMAL transition must be enabled when intrinsic reporting is to be used; this shall be set by default. This property is required if intrinsic reporting is supported by this object.

12.25.23 Acked_Transitions

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

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