13.11 GetEnrollmentSummary Service

The GetEnrollmentSummary service is used by a client BACnet-user to obtain a summary of event-initiating objects. Several different filters may be applied to define the search criteria. This service may be used to obtain summaries of objects with any EventType and is thus a superset of the functionality provided by the GetAlarmSummary Service.

13.11.1 Structure

The structure of the GetEnrollmentSummary service primitives is shown in Table 13-12. The terminology and symbology used in this table are explained in 5.6.

Table 13-12. Structure of GetEnrollmentSummary Service Primitives

Parameter Name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Acknowledgment Filter	M	M(=)		
Enrollment Filter	U	U(=)		
Event State Filter	U	U(=)		
Event Type Filter	U	U(=)		
Priority Filter	U	U(=)		
Notification Class Filter	U	U(=)		
Result(+)			S	S(=)
List of Enrollment Summaries			M	M(=)
Object Identifier			M	M(=)
Event Type			M	M(=)
Event State			M	M(=)
Priority			M	M(=)
Notification Class			U	U(=)
Result(-)			S	S(=)
Error type			M	M(=)

13.11.1.1 Argument

This parameter shall convey the parameters for the GetEnrollmentSummary confirmed service request.

13.11.1.1.1 Acknowledgment Filter

This parameter, of type ENUMERATED, shall provide a means of restricting the event-initiating objects that are to be summarized. The 'Acknowledgment Filter' may take any of three values:

> ALL - Shall request that the returned summary contain all event-initiating objects without regard to whether the objects have acknowledgments or not.
>
> ACKED - Shall request that the returned summary contain only those objects for which the Acked_Transitions property has a value of one in every bit position.
>
> NOT-ACKED - Shall request that the returned summary contain only reports for those objects for which the Acked_Transitions property has a value of zero in one or more bit positions.

13.11.1.1.2 Enrollment Filter

This parameter, of type BACnetRecipientProcess, shall provide a means of restricting the set of objects that are to be summarized. Only those objects for which the specified BACnetRecipient and Process Identifier are enrolled to receive notifications, either confirmed or unconfirmed, shall be summarized. In this case, "enrolled" shall mean that an event-initiating object references a Notification Class object containing one or more BACnetDestinations containing the indicated Process Identifier and BACnetRecipient.

If this parameter is omitted, it shall mean that event-initiating objects shall be summarized without regard to enrollment status.

13.11.1.1.3 Event State Filter

This parameter shall provide a means of restricting the set of event-initiating objects that are to be summarized. It may have any of the following values:

{OFFNORMAL, FAULT, NORMAL, ALL, ACTIVE}.

Only those event-initiating objects whose Event_State property matches the value specified in this parameter shall be included. If the value ALL is specified, then all of the event-initiating objects shall be summarized without regard to the value of the Event_State property. If the value ACTIVE is specified, then only those event-initiating objects whose Event_State property has a value other than NORMAL shall be summarized. If this parameter is omitted, a default value of ALL shall be assumed.

13.11.1.1.4 Event Type Filter

This parameter is provided as a means of restricting the summary to only those event-initiating objects that can generate event notifications with an Event_Type equal to the value of this parameter. This parameter may have any legal value of Event_Type as defined in the Event Enrollment object specification. If this parameter is omitted, all event-initiating objects shall be included in the summary without regard to which EventTypes they generate.

13.11.1.1.5 Priority Filter

This parameter consists of two parts, MinPriority and MaxPriority, each of datatype Unsigned8. It provides a means of restricting the summary to only those event-initiating objects that can generate event notifications with a Priority as specified by this parameter. The 'Priority Filter' parameter consists of two parts, MinPriority and MaxPriority. All event-initiating objects, such that MinPriority ≤ Priority ≤ MaxPriority, shall be included in the summary. If 'Priority Filter' is omitted, all event-initiating objects shall be summarized without regard to their Priority.

13.11.1.1.6 Notification Class Filter

This parameter, of type Unsigned, provides a means of restricting the summary to only those event-initiating objects that can generate event notifications with a Notification Class equal to the value of this parameter. If 'Notification Class Filter' is omitted, it shall mean that all event-initiating objects objects shall be summarized without regard to their Notification Class.

13.11.1.2 Result(+)

The 'Result(+)' parameter shall indicate that the requested service has succeeded. A successful result includes the following parameters.

13.11.1.2.1 List of Enrollment Summaries

The 'List of Enrollment Summaries' shall contain zero or more Enrollment Summaries. Each Enrollment Summary shall consist of up to five parameters: 'Object Identifier', 'Event Type', 'Event State', 'Priority', and, optionally, 'Notification Class'. If the list is of length zero, then this shall be interpreted to mean that there are no event-initiating objects that meet the search criteria specified in the request primitive.

13.11.1.2.1.1 Object Identifier

This parameter, of type BACnetObjectIdentifier, shall identify an object meeting the search criteria.

13.11.1.2.1.2 Event Type

This parameter, of type BACnetEventType, indicates the Event_Type that the object can generate.

13.11.1.2.1.3 Event State

This parameter, of type BACnetEventState, indicates the value of the Event_State property of the object.

13.11.1.2.1.4 Priority

This parameter, of type Unsigned8, indicates the priority of notifications generated by the object.

13.11.1.2.1.5 Notification Class

This optional parameter, of type Unsigned, indicates the class of notifications generated by the object and implicitly refers to a Notification Class object that has a Notification_Class property of the same value.

13.11.1.3 Result(-)

The 'Result(-)' parameter shall indicate that the service request has failed. The reason for failure shall be specified by the 'Error Type' parameter.

13.11.1.3.1 Error Type

This parameter shall consist of two component parameters: (1) the 'Error Class' and (2) the 'Error Code'. See Clause 18.

13.11.2 Service Procedure

After verifying the validity of the request, the responding BACnet-user shall search for all event-initiating objects that meet the search criteria specified in the request primitive. The search criteria are the logical conjunctions of all of the explicitly stated filters and the default values for any filters that were omitted in the request primitive. A positive response containing the enrollment summaries for objects found in this search shall be constructed. If no objects are found that meet these criteria, then a list of length zero shall be returned.

13.12 GetEventInformation Service

The GetEventInformation service is used by a client BACnet-user to obtain a summary of all "active event states". The term "active event states" refers to all event-initiating objects that

(a) have an Event_State property whose value is not equal to NORMAL, or
(b) have an Acked_Transitions property, which has at least one of the bits (TO-OFFNORMAL, TO-FAULT, TO-NORMAL) set to FALSE.

This service is intended to be implemented in all devices that generate event notifications.

13.12.1 Structure

The structure of the GetEventInformation service primitives is shown in Table 13-13. The terminology and symbology used in this table are explained in 5.6.

Table 13-13. Structure of GetEventInformation Service Primitives

Parameter Name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Last Received Object Identifier	U	U(=)		
Result(+)				
List of Event Summaries			M	M(=)
Object Identifier			M	M(=)
Event State			M	M(=)
Acknowledged Transitions			M	M(=)
Event Time Stamps			M	M(=)
Notify Type			M	M(=)
Event Enable			M	M(=)
Event Priorities			M	M(=)
More Events			M	M(=)
Result(-)			S	S(=)
Error Type			M	M(=)

13.12.1.1 Argument

This parameter indicates the GetEventInformation confirmed service request.

13.12.1.1.1 Last Received Object Identifier

This optional parameter, of type BACnetObjectIdentifier, shall specify the last Object Identifier received in a preceding GetEventInformation-ACK, if its 'More Events' parameter was TRUE. If this parameter is omitted, the returned summary shall start with the first object meeting the "active event states" criteria. A fixed object processing order is assumed, however the particular order is a local matter. If the Last Received Object Identifier has become invalid in the responding device (i.e., the object is no longer present), the service shall resume if it is possible to determine the object that would have been the successor of the object that is no longer present. Otherwise a Result(-) shall be returned with an error class of OBJECT and an error code of UNKNOWN_OBJECT.

13.12.1.2 Result(+)

The 'Result(+)' parameter shall indicate that the requested service has succeeded. A successful result includes the following parameters.

13.12.1.2.1 List of Event Summaries

The 'List of Event Summaries' shall contain zero or more Event Summaries. Each Event Summary shall consist of seven parameters: 'Object Identifier', 'Event State', 'Acknowledged Transitions', 'Event Time Stamps', 'Notify Type', 'Event Enable' and 'Event Priorities'. If the list is of length zero, then this shall be interpreted to mean that there are no event-initiating objects that have active event states in this device.

13.12.1.2.1.1 Object Identifier

This parameter, of type BACnetObjectIdentifier, shall identify the event-initiating object that has an Event_State property whose value is not equal to NORMAL or has an Acked_Transitions property that has at least one of the following bits (TO-OFFNORMAL, TO-FAULT, TO-NORMAL) set to FALSE.

13.12.1.2.1.2 Event State

This parameter, of type BACnetEventState, indicates the value of the Event_State property of the object.

13.12.1.2.1.3 Acknowledged Transitions

This parameter, of type BACnetEventTransitionBits, indicates the value of the Acked_Transitions property of the object.

13.12.1.2.1.4 Event Time Stamps

This parameter, of type BACnetARRAY[3] of BACnetTimeStamp, shall convey the timestamps of the last event notifications for TO-OFFNORMAL, TO-FAULT, and TO-NORMAL events.

13.12.1.2.1.5 Notify Type

This parameter, of type BACnetNotifyType, shall convey the value of the Notify_Type property of this object.

13.12.1.2.1.6 Event Enable

This parameter, of type BACnetEventTransitionBits, shall convey the value of the Event_Enable property of the object.

13.12.1.2.1.7 Event Priorities

This parameter, of type BACnetARRAY[3] of Unsigned, shall convey the priorities specified in the Priority property of the associated Notification Class object. In the case where an Event Enrollment Object is used without an associated Notification Class Object, the three fields of this parameter shall all contain the value of the Priority property of the Event Enrollment Object.

13.12.1.2.2 More Events

This parameter, of type BOOLEAN, shall indicate whether (TRUE) or not (FALSE) more objects exist that meet the active event state criteria of the service request, but that could not be returned in the reply.

13.12.1.3 Result(-)

The 'Result(-)' parameter shall indicate that the service request has failed. The reason for failure shall be specified by the 'Error Type' parameter.

13.12.1.3.1 Error Type

This parameter shall consist of two component parameters: (1) the 'Error Class' and (2) the 'Error Code'. See Clause 18.

13.12.2 Service Procedure

After verifying the validity of the request, the responding BACnet-user shall search for all event-initiating objects that meet the following conditions, beginning with the object following (in whatever internal ordering of objects is used by the responding device) the object specified by the 'Last Received Object Identifier' parameter, if present:

(a) have an Event_State property whose value is not equal to NORMAL, or
(b) have an Acked_Transitions property that has at least one of the following bits (TO-OFFNORMAL, TO-FAULT, TO-NORMAL) set to FALSE.

A positive response containing the event summaries for objects found in this search shall be constructed. If no objects are found that meet these criteria, then a list of length zero shall be returned. As many of the included objects as can be returned within the APDU shall be returned. If more objects exist that meet the criteria but cannot be returned in the APDU, the 'More Events' parameter shall be set to TRUE, otherwise it shall be set to FALSE.

13.13 LifeSafetyOperation Service

The LifeSafetyOperation service is intended for use in fire, life safety and security systems to provide a mechanism for conveying specific instructions from a human operator to accomplish any of the following objectives:

(a) silence audible or visual notification appliances,
(b) reset latched notification appliances, or
(c) unsilence previously silenced audible or visual notification appliances.

Ensuring that the LifeSafetyOperation request actually comes from a person with appropriate authority is a local matter.

13.13.1 Structure

The structure of the LifeSafetyOperation primitive is shown in Table 13-14. The terminology and symbology used in this table are explained in 5.6.

Table 13-14. Structure of LifeSafetyOperation Service Primitives

Parameter Name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Requesting Process Identifier	M	M(=)		
Requesting Source	M	M(=)		
Request	M	M(=)		
Object Identifier	U	U(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)

13.13.1.1 Argument

This parameter shall convey the parameters for the LifeSafetyOperation confirmed service request.

13.13.1.1.1 Requesting Process Identifier

This parameter, of type Unsigned32, specifies an identifying number of significance to the sending device that uniquely identifies the process which initiated the service request. The assignment and meaning of process identifiers shall be a local matter.

13.13.1.1.2 Requesting Source

This parameter, of type CharacterString, specifies the identity of the human operator that initiated the LifeSafetyOperation service request.

13.13.1.1.3 Request

This parameter, of type BACnetLifeSafetyOperation, shall convey the requested operation:

{SILENCE, SILENCE_AUDIBLE, SILENCE_VISUAL, RESET, RESET_ALARM, RESET_FAULT, UNSILENCE, UNSILENCE_AUDIBLE, UNSILENCE_VISUAL}

13.13.1.1.4 Object Identifier

This parameter, of type BACnetObjectIdentifier, shall convey the specific BACnet object to which the life safety request is directed. If this parameter is not present, then all applicable objects within the receiving BACnet device shall be silenced or reset accordingly based on the 'Request' provided.

13.13.1.2 Result(+)

The 'Result(+)' parameter shall indicate that the service request succeeded.

13.13.1.3 Result(-)

The 'Result(-)' parameter shall indicate that the service request has failed. The reason for the failure shall be specified by the 'Error Type' parameter.

13.13.1.3.1 Error Type

This parameter consists of two component parameters: (1) the 'Error Class' and (2) the 'Error Code'. See Clause 18.

13.13.2 Service Procedure

The responding BACnet-user shall first verify the validity of the 'Object Identifier' parameter and return a 'Result(-)' response with the appropriate error class and code if the 'Request' is invalid or if the 'Object Identifier' parameter is present and specifies an object that is either unknown or does not represent an appropriate request for this object type.

If the 'Object Identifier' parameter is not present, then the responding BACnet-user shall attempt to operate all applicable objects in the device based on the 'Request' parameter. If the 'Object Identifier' parameter is present, the responding BACnet-user shall attempt to silence or reset the object specified in the 'Object Identifier' parameter based on the 'Request' parameter. In either case, the responding BACnet-user shall issue a Result(+) primitive.

13.14 SubscribeCOV Service

The SubscribeCOV service is used by a COV-client to subscribe for the receipt of notifications of changes that may occur to the properties of a particular object. Certain BACnet standard objects may optionally support COV reporting. If a standard object provides COV reporting, then changes of value of specific properties of the object, in some cases based on programmable increments, trigger COV notifications to be sent to one or more subscriber clients. Typically, COV notifications are sent to supervisory programs in BACnet client devices or to operators or logging devices. Proprietary objects may support COV reporting at the implementor's option. The standardized objects that may optionally provide COV support and the change of value algorithms they shall employ are summarized in Table 13-1.

The subscription establishes a connection between the change of value detection and reporting mechanism within the COV-server device and a "process" within the COV-client device. Notifications of changes are issued by the COV-server device when changes occur after the subscription has been established. The ConfirmedCOVNotification and UnconfirmedCOVNotification services are used by the COV-server device to convey change notifications. The choice of confirmed or unconfirmed service is made at the time the subscription is established.

13.14.1 Structure

The structure of the SubscribeCOV service primitives is shown in Table 13-15. The terminology and symbology used in this table are explained in 5.6.

Table 13-15. Structure of SubscribeCOV Service Primitives

Parameter Name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Subscriber Process Identifier	M	M(=)		
Monitored Object Identifier	M	M(=)		
Issue Confirmed Notifications	U	U(=)		
Lifetime	U	U(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)

13.14.1.1 Argument

This parameter shall convey the parameters for the SubscribeCOV confirmed service request.

13.14.1.2 Subscriber Process Identifier

This parameter, of type Unsigned32, shall convey a numeric "handle" meaningful to the subscriber. This handle shall be used to match future re-subscriptions and cancellations from the subscriber with the COV context that exists within the COV-server device and with confirmed or unconfirmed COV notifications to identify the process within the COV-client that should receive them. The value zero is reserved for unsubscribed COV notifications as described in 13.7.

13.14.1.3 Monitored Object Identifier

This parameter, of type BACnetObjectIdentifier, shall convey the identifier of the object within the receiving device for which a subscription is desired.

13.14.1.4 Issue Confirmed Notifications

This parameter, of type BOOLEAN, shall convey whether the COV-server device shall issue ConfirmedCOVNotifications (TRUE) or UnconfirmedCOVNotifications (FALSE) when changes occur. This parameter, if present, shall indicate a subscription or re-subscription is to occur and that the lifetime shall be refreshed to its initial state. If both the 'Issue Confirmed Notifications' and 'Lifetime' parameters are absent, then this shall indicate a cancellation request. If the 'Lifetime' parameter is present then the 'Issue Confirmed Notifications' parameter shall be present.

13.14.1.5 Lifetime

This parameter, of type Unsigned, shall convey the desired lifetime of the subscription in seconds. A value of zero shall indicate an indefinite lifetime, without automatic cancellation. A non-zero value shall indicate the number of seconds that may elapse before the subscription shall be automatically cancelled. If both the 'Issue Confirmed Notifications' and 'Lifetime' parameters are absent, then this shall indicate a cancellation request. If the 'Lifetime' parameter is present then the 'Issue Confirmed Notifications' parameter shall be present.

13.14.1.6 Result(+)

The 'Result(+)' parameter shall indicate that the requested service has succeeded.

13.14.1.7 Result(-)

The 'Result(-)' parameter shall indicate that the service request has failed. The reason for failure shall be specified by the 'Error Type' parameter.

13.14.1.7.1 Error Type

This parameter shall consist of two component parameters: (1) the 'Error Class' and (2) the 'Error Code'. See Clause 18.

13.14.2 Service Procedure

If neither 'Lifetime' nor 'Issue Confirmed Notifications' are present, then the request shall be considered to be a cancellation. Any COV context that already exists for the same BACnet address contained in the PDU that carries the SubscribeCOV request and has the same 'Subscriber Process Identifier' and 'Monitored Object Identifier' shall be disabled and a 'Result(+)' returned. Cancellations that are issued for which no matching COV context can be found shall succeed as if a context had existed, returning 'Result(+)'.

If the 'Lifetime' parameter is present and has a non-zero value but the device does not support automatic cancellation of subscriptions, then a 'Result(-)' shall be returned. If the 'Lifetime' parameter is not present but the 'Issue Confirmed Notifications' parameter is present, then a value of zero (indefinite lifetime) shall be assumed for the lifetime. If the 'Issue Confirmed Notifications' parameter is present but the object to be monitored does not support COV reporting, then a 'Result(-)' shall be returned. If the object to be monitored does support COV reporting, then a check shall be made to locate an existing COV context for the same BACnet address contained in the PDU that carries the SubscribeCOV request and has the same 'Subscriber Process Identifier' and 'Monitored Object Identifier'. If an existing COV context is found, then the request shall be considered a re-subscription and shall succeed as if the subscription had been newly created.

If no COV context can be found that matches the request, then a new COV context shall be established that contains the BACnet address from the PDU that carries the SubscribeCOV request and the same 'Subscriber Process Identifier' and 'Monitored Object Identifier'. If no context can be created, then a 'Result(-)' shall be returned.

If a new context is created, or a re-subscription is received, then the COV context shall be initialized and given a lifetime as specified by the 'Lifetime' parameter, if present, or zero if the 'Lifetime' parameter is not present. The subscription shall be automatically cancelled after that many seconds have elapsed unless a re-subscription is received. A lifetime of zero shall indicate that the subscription is indefinite and no automatic cancellation shall occur. In either case, a 'Result(+)' shall be returned. A ConfirmedCOVNotification or UnconfirmedCOVNotification shall be issued as soon as possible after the successful completion of a subscription or re-subscription request, as specified by the 'Issue Confirmed Notifications' parameter.

13.15 SubscribeCOVProperty Service

The SubscribeCOVProperty service is used by a COV-client to subscribe for the receipt of notifications of changes that may occur to the properties of a particular object. Any object may optionally support COV reporting. If a standard object provides COV reporting, then changes of value of subscribed-to properties of the object, in some cases based on programmable increments, trigger COV notifications to be sent to one or more subscriber clients. Typically, COV notifications are sent to supervisory programs in BACnet client devices or to operators or logging devices.

The subscription establishes a connection between the change of value detection and reporting mechanism within the COV-server device and a "process" within the COV-client device. Notifications of changes are issued by the COV-server device when changes occur after the subscription has been established. The ConfirmedCOVNotification and UnconfirmedCOVNotification services are used by the COV-server device to convey change notifications. The choice of confirmed or unconfirmed service is made at the time the subscription is established. Any object, proprietary or standard, may support COV reporting for any property at the implementor's option.

The SubscribeCOVProperty service differs from the SubscribeCOV service in that it allows monitoring of properties other than those listed in Table 13-1.

13.15.1 Structure

The structure of the SubscribeCOVProperty service primitives is shown in Table 13-16. The terminology and symbology used in this table are explained in 5.6.

Table 13-16. Structure of SubscribeCOVProperty Service Primitives

Parameter Name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
Subscriber Process Identifier	M	M(=)		
Monitored Object Identifier	M	M(=)		
Issue Confirmed Notifications	U	U(=)		
Lifetime	U	U(=)		
Monitored Property Identifier	M	M(=)		
COV Increment	U	U(=)		
Result(+)			S	S(=)
Result(-)			S	S(=)
Error Type			M	M(=)

13.15.1.1 Argument

This parameter shall convey the parameters for the SubscribeCOVProperty confirmed service request.

13.15.1.2 Subscriber Process Identifier

This parameter, of type Unsigned32, shall convey a numeric "handle" meaningful to the subscriber. This handle shall be used to match future re-subscriptions and cancellations from the subscriber with the COV context that exists within the COV-server device and with confirmed or unconfirmed COV notifications to identify the process within the COV-client that should receive them.

13.15.1.3 Monitored Object Identifier

This parameter, of type BACnetObjectIdentifier, shall convey the identifier of the object within the receiving device that contains the property for which a subscription is desired.

13.15.1.4 Issue Confirmed Notifications

This parameter, of type BOOLEAN, shall convey whether the COV-server device shall issue ConfirmedCOVNotifications (TRUE) or UnconfirmedCOVNotifications (FALSE) when changes occur. This parameter, if present, shall indicate that a subscription or re-subscription is to occur and that the lifetime shall be refreshed to its initial state. If both the 'Issue