



# Platform Services

## BACnet Alarm Management



### APPLICATION NOTE

August 2014

**Description:** Guide to understanding, setting up, and processing BACnet Alarms.

**General Requirement:** Basic GENESIS64 and BACnet knowledge.

## Introduction

BACnet protocol itself offers all-in-one Alarm management features including algorithmic-based generation of real-time alarms (similar to AlarmWorX64 Server), logging these alarms to internal buffer (similar to AlarmWorX64 Logger). It is important to realize that all operations regarding BACnet Alarms are running inside a BACnet device. GENESIS64 acts here as a client which configures properties of individual objects, as well as a client which is capable to visualize (and do further processing) of real-time and historical alarms.

## BACnet Alarms

It is important to know about BACnet alarms that there's an object type called Notification Class. Instance of this object type is used as a 'gateway' for reporting about new real-time alarms. Client (in our case AlarmWorX64 Viewer or Logger via BACnet Runtime) subscribes to a selected Notification Class. When a new event occurs the Notification Class sends a notification to all subscribed clients. BACnet device can contain almost any amount of Notification Class objects.

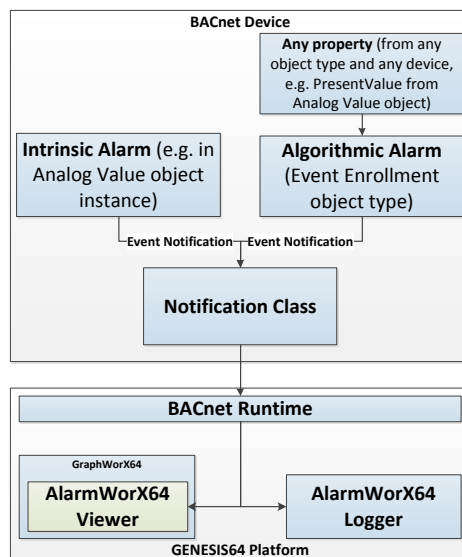


Figure 1 - BACnet Alarms/Event Infrastructure

## Algorithmic (Real-time) Alarms – Event Enrollment object type

Event Enrollment is an object type allowing you to reference any property from any object instance, located in any BACnet device as a datasource. Then it allows you to choose an algorithm for evaluating alarm conditions. If these specific conditions are met, then event notification is sent to a chosen Notification Class Object. Then it is distributed to subscribers.

## Intrinsic (Real-time) Alarms

Some object types (Analog Input/Output/Value, Binary Input/Output/Value) supports intrinsic alarming – it means that the object instance can be configured to generate event notifications based on object-specific rules (Analog object types are based on Low, Normal and High Limit notifications, Binary object types can create event notifications based on Binary object state) – it is not required to connect a property with an Event Enrollment.

## Historical Alarms – Event Log object type

If a BACnet device supports Event Logs, it is capable to store specific amount of BACnet events from a Notification Class. AlarmWorX64 Viewer can be then used for visualization and navigation through the buffer.

## Event States

In BACnet universe there are 3 standardized event states:

- **Normal** – value is within specified boundaries
- **Off Normal** – value is out of specified boundaries
- **Fault** – a problem prevents the device from providing correct event evaluation (e.g. sensor malfunctioned, thus cannot reliably provide a value)

## Creating Notification Class

In order to create a Notification Class please follow these steps:

1. Launch Workbench-SL.
2. Switch to the BACnet provider.
3. Find the desired device in the object tree, right-click it and select the 'Add Object' item.
4. Choose 'Notification Class' in the drop-down list.
5. By default a form will be opened where you should define an instance ID, state priorities, and whether acknowledgement is required. You can also specify a list of recipients in Recipient List pane – by this you will explicitly tell which BACnet devices should receive notifications. It substitutes subscription from the device. Click 'Apply' to create a Notification Class.



# Platform Services

## BACnet Alarm Management



### APPLICATION NOTE

August 2014

A device can contain multiple Notification Classes. If you are familiar with the concept of AlarmWorX64 Server, the Notification Classes can be compared to Areas – they can be used to distinguish alarms/events from specific set of alarm/event sources. You can subscribe to a notification class of a subset of alarms/events only. For instance imagine that you have two rooms, each with a set of BACnet-enabled sensors and detectors (temperature, humidity, light intensity, etc.), all these sensors are connected to a single central BACnet device. You will define that alarms/events from sensors located in Room A will notify through Notification Class (NC) with instance 0 and sensors located in Room B through NC 1. This will allow you to separate and clearly distinguish events raised in separate rooms.

### Setting up intrinsic alarm

In order to have intrinsic alarming working as expected it is required to have a Notification Class already created in the device (you have to know its instance number). Follow the next steps to set it up:

**NOTE:** Not all objects are intrinsic alarming-enabled.

1. Launch Workbench Silverlight.
2. Switch to the BACnet provider.
3. Find an object supporting intrinsic alarming and select it e.g. an analog value provided by ICONICS BACnet Simulator located on a remote machine.
4. The Runtime tab of the selected object will be opened and you should be able to read and write values of all properties. You have to write specific values to the following properties (under BACnet / Intrinsic Alarming section):
  - a. **Notification Class** – instance number of the NC you want to use for alarm notifications
  - b. **Time Delay** – how long (in seconds) the value must stay over/under the specified limit before an Event is generated (for instant notification set to 0)
  - c. **Event Enable** – which event transitions will be notified (check at least To Normal and To Off Normal events)
  - d. **Limit Enable** – this property is Analog object type specific – specify which limit should be monitored in order to throw events (in our case check both – High Limit Enable and Low Limit Enable)
  - e. **High Limit** – specify value of High Limit (set 300)
  - f. **Low Limit** – specify value of Low Limit (set 100)

Please note that other Object Types can have different Intrinsic Alarm settings depending on object's purpose (e.g. Binary object types will not have Low and High Limits). But Notification Class, Time Delay, and Event Enable properties should be used for all.

### Setting up algorithmic alarm

In order to have algorithmic alarming working as expected it is required to have a Notification Class and the monitored property already created in the device. Algorithmic alarming uses Event Enrollment object type to configure alarm/event source property, method – algorithm used for the alarm/event evaluation, and Notification Class as alarm/event 'reporter'. To set it up please follow these steps:

1. Launch Workbench Silverlight.
2. Switch to the BACnet provider.
3. Find the desired device in the tree,
4. Right-click it and select the 'Add Object' item.
5. Choose 'Event Enrollment' in the drop-down list and confirm by clicking OK (don't have 'Create Empty Object' checkbox checked).
6. In opened form you have to fill in following:
  - a. **Notification Class** – instance number of NC you want to use for alarm notifications.
  - b. **Event Type** – here you choose an algorithm type used for evaluation. Please refer to BACnet specification for details (you can find brief information about specific Event Types in Figure 2) – in our case please set 'Out Of Range' Event Type
  - c. **Event Parameters** – Click '...' button next to 'Event Parameters' text box to open the dialog specific for 'Out Of Range' event type in order to set its parameters – it uses same parameters as we described in 'Setting up intrinsic alarm' section so please refer it for the details.
  - d. **Referenced Object** – Click '...' button next to 'Referenced Object' text box to open the dialog to browse for referenced property whose value is used as a data source, or set Referenced Object values manually using combo boxes and numeric fields.
7. Confirm Event Enrollment object creation by clicking Apply button.



# Platform Services

## BACnet Alarm Management



APPLICATION NOTE

August 2014

| Event type            | Event State                               | Event Parameters                                                                                  |
|-----------------------|-------------------------------------------|---------------------------------------------------------------------------------------------------|
| Change Of Bitstring   | Normal<br>Off Normal                      | Time Delay<br>Bitmask<br>List of Bitstring Values                                                 |
| Change Of State       | Normal<br>Off Normal                      | Time Delay<br>List Of Values                                                                      |
| Change Of Value       | Normal<br>Off Normal                      | Time Delay<br>Bitmask<br>Referenced<br>Property Increment                                         |
| Command Failure       | Normal<br>Off Normal                      | Time Delay<br>Feedback Property Reference                                                         |
| Floating Limit        | Normal<br>High Limit<br>Low Limit         | Time Delay<br>Setpoint Reference<br>Low Diff Limit<br>High Diff Limit<br>Deadband                 |
| Out Of Range          | Normal<br>High Limit<br>Low Limit         | Time Delay<br>Low Limit<br>High Limit<br>Deadband                                                 |
| Unsigned Range        | Normal<br>High Limit<br>Low Limit         | Time Delay<br>Low Limit<br>High Limit                                                             |
| Change Of Life Safety | Normal<br>Off Normal<br>Life Safety Alarm | Time Delay<br>List Of Alarm Values<br>List Of Life Safety Alarm Values<br>Mode Property Reference |

Figure 2 - Event Type list

### Before you subscribe to Notification Class object

To be able to see or log BACnet Alarms from a device you have to put interested devices into Alarm cache before your client can get any ICONICS BACnet Alarms. You can add devices to alarm cache during Device Discovery process by clicking on buttons with alarm indicator as shown in Figure 3.

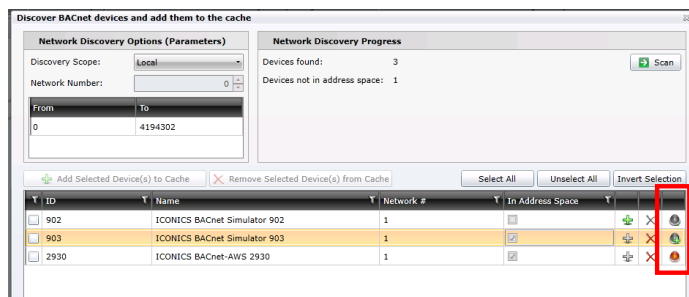


Figure 3 - Subscribing to alarms during device discovery

Another option how to subscribe for BACnet Alarms from a device is right-clicking on already added device in the tree of Workbench Silverlight BACnet provider and choose 'Add to Alarm Cache'.

### Subscribing AlarmWorX64 Viewer to Notification Class object (Real-Time alarms)

In order to display BACnet Alarms in AlarmWorX64 Viewer hosted in GraphWorX64, do the following:

1. Launch GraphWorX64.
2. Insert an AlarmWorX64 Viewer control
3. Double-click the viewer control in order to open its properties and settings.

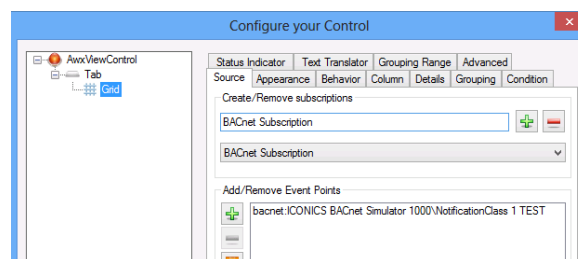


Figure 4 - Configuring BACnet Alarm Views

4. In the opened dialog, select the 'Grid' tree item.
5. Click the button with the '+' sign in the 'Create/Remove Subscriptions' section and choose 'Real Time Subscription' from the opened context menu.
6. Select the 'Local Alarm Server' item in the 'Add/Remove Event Points' section and click the '-' sign in the same section to remove the existing subscription.
7. Now press the '+' button in the 'Add/Remove Event Points' section. This will open the Data Browser.
8. In the Data Browser's 'Shortcuts' pane, on the left side, choose 'Home' and you'll be switched to the root of Data Browser.
9. In the 'Grid', choose 'BACnet' at the very top of the list



# Platform Services

## BACnet Alarm Management



### APPLICATION NOTE

August 2014

10. Choose your BACnet device from the list to see its provided objects.
11. In the list of objects, find a Notification Class providing notifications about alarms. These objects are marked with the attribute 'Subscribe', which is visible in the 'Attribute' column. You can sort the list by clicking the column header.
12. Confirm selection by clicking 'OK' in the Data Browser. This will add the Notification Class subscription to the AlarmWorX64 Viewer.
13. After subscribing to all desired Notification Classes (repeat steps 7 to 12 to subscribe to more NCs), you can close the dialog window for AlarmWorX64 Viewer's configuration.

**NOTE:** You can also subscribe to a device or the complete BACnet network in the Data Browser. In this way you can subscribe to all NC in the device or all BACnet devices in the alarm cache.

14. Now switch GraphWorX64 to Runtime mode in order to receive and see alarms provided by your subscribed Notification Classes.

### Subscribing AlarmWorX64 Viewer to Event Log object (Historical alarms)

In order to display BACnet Historical Alarms (Event Logs) in AlarmWorX64 Viewer hosted in GraphWorX64, do the following:

15. Launch GraphWorX64.
16. Insert an AlarmWorX64 Viewer control.
17. Double-click the viewer control in order to open its properties and settings.
18. In the opened dialog, select the 'Grid' tree item.
19. Now click the button with the '+' sign in the 'Create/Remove Subscriptions' section and choose 'Historical Subscription' from the opened context menu.
20. Select the 'Local Alarm Server' list box item in the 'Add/Remove Event Points' section and click the '-' sign in the same section to remove the existing subscription.
21. Press '+' button in the 'Add/Remove Event Points' section. This will open the Data Browser.

22. In the Data Browser's 'Shortcuts' pane, on the left side, choose 'Home' and you'll be switched to the root of the Data Browser.
23. In the 'Grid', choose 'BACnet' at the very top of the list.
24. Choose your BACnet device from the list to see its provided objects.
25. In the list of objects, find an Event Log object providing historical alarms. These objects are marked with the attribute 'HistRead', which is visible in the 'Attribute' column. You can sort the list by clicking the column header.
26. Confirm your selection by clicking 'OK' in the Data Browser. This will add the Event Log historical subscription to the AlarmWorX64 Viewer.
27. After listing all desired Event Log objects (repeat steps 7 to 12), you can close the dialog window for AlarmWorX64 Viewer's configuration.
28. Switch GraphWorX64 to Runtime mode in order to see historical alarms provided by the Event Log object(s).

**NOTE:** Current implementation supports getting the historical data by polling method only; AlarmWorX64 Viewer asks for new data. This can be forced manually while GraphWorX64 with an AlarmWorX64 Viewer control is in Runtime mode and the control is 'Refreshed'. Or, you can enable automatic polling in AlarmWorX64 Viewer's configuration dialog: via the 'Behavior' tab, click the checkbox and fill in the input field 'Automatically update Historical data after'.

### Logging BACnet Alarms to SQL Database

You can log BACnet Alarms using AlarmWorX64 Logger. You just need to select a Notification Class object instance as a point for Alarm Subscription.

### Setting up AlarmWorX64 Server to generate alarms on BACnet properties

It is possible to have AlarmWorX64 Server trigger alarms based on values coming from BACnet devices. The process of setting up the rules for triggering an alarm is the same as in case of an OPC DA alarm but you select a BACnet property for the alarm data source.