



BACnet

Vergleich Intrinsic und Algorithmic Reporting - Die Sicht des Projektengineurs -

1. EINFÜHRUNG	3
1.1. HINTERGRUND	3
1.2. MARKT SITUATION	3
1.3. ALLGEMEINE BESCHREIBUNG.....	4
1.4. VERWENDETE ABBILDUNGEN	5
1.5. LAST BUT NOT LEAST	5
2. BESCHREIBUNG DES ENGINEERING	6
2.1. ALLGEMEINES.....	6
2.2. INTRINSIC REPORTING	7
2.3. ALGORITHMIC REPORTING.....	9
3. VERGLEICH	14
3.1. INGENIEURTECHNISCHE BEARBEITUNG	14
3.2. BACNET OBERFLÄCHE FÜR BENUTZER / BEDIENER	16
3.3. ZUSAMMENFASSUNG.....	16
4. ANDERE VERÖFFENTLICHUNGEN IM ZUSAMMENHANG MIT BEIDEN METHODEN DER EREIGNISBENACHRICHTIGUNG.....	17
5. LITERATUR.....	17

1. Einführung

1.1. Hintergrund

Bei Installationen der Gebäudeautomation ist das Alarm Management eine zentrale und wichtige Funktion. Der BACnet Standard definiert zwei Methoden um diese Funktionalität zu erfüllen, *Algorithmic Reporting* und *Intrinsic Reporting*.

Oftmals fordern Fachplaner in Ausschreibungen für BACnet Projekte **entweder** *Algorithmic Reporting* **oder** *Intrinsic Reporting* zur Erfüllung der Aufgaben des Alarm Managements und schließen ausdrücklich die jeweils andere Methode der Ereignisbenachrichtigung aus. Die Gründe dieses Verhaltens können z.B. sein:

- (1) Es besteht nicht genug Wissen über die praktischen Aspekte der beiden Methoden der Ereignisbenachrichtigung.
- (2) Es ist eine Möglichkeit bestimmte Hersteller zu bevorzugen und die Produkte anderer Hersteller auszuschließen.

Gegen den Grund (1) kann dieses Dokument hoffentlich helfen.

Das Hauptanliegen dieses Dokumentes ist die praktische Anwendung beider Methoden des Alarm Managements. Das grundsätzliche Wissen über das Alarm Management wird vorausgesetzt.

Weiterhin gab es Überlegungen, den BACnet Standard zu erweitern, da mit *Intrinsic Reporting* für analoge Datenpunkte nur die Überwachung mit einem Grenzwertpaar möglich ist, häufig jedoch 2 Grenzwertpaare benötigt werden. Der BACnet Standard muss aus diesem Grunde allerdings nicht erweitert werden, da dieses Thema mit *Algorithmic Reporting* lösbar ist (siehe nachfolgende Beispiele).

Einzelheiten über *Algorithmic Reporting* und *Intrinsic Reporting* sind in [1] und [2] beschrieben (Englisch). Weitere Dokumente in Deutscher Sprache sind: [3], [4] und [5].

1.2. Markt Situation

Bezüglich des Alarm Managements können verfügbare BACnet Geräte (devices) wie folgt klassifiziert werden:

- einfache Geräte ohne Alarm Management
- Geräte mit *Algorithmic Reporting*
- Geräte mit *Intrinsic Reporting*
- Geräte mit *Algorithmic Reporting* und *Intrinsic Reporting*

Wenn in einem realen Projekt unterschiedliche Typen von Geräten verwendet werden, macht es keinen Sinn zu fordern, dass von allen Geräten eine oder beide Methoden des Alarm Managements unterstützt werden müssen. Wenn es eine große Anzahl von Geräten ohne eigenes Alarm Management (z.B. Raumregler mit fester Applikation, Frequenzumrichter o.ä.)

gibt, kann das Alarm Management für diese Geräte durch andere Geräte mit eigenem Alarm Management wahrgenommen werden.

1.3. Allgemeine Beschreibung

Zur Durchführung eines Alarm Managements in Systemen der Gebäudeautomation sind 3 Aspekte wichtig:

- (1) Welcher Datenpunkt soll verantwortlich sein für die Auslösung eines Alarms?
- (2) In welcher Situation soll dieser Alarm ausgelöst werden?
- (3) Zu wem soll dieser Alarm gesendet werden?

Beim *Intrinsic Reporting* ist Aspekt (1) identisch mit dem Datenpunkt selbst während Aspekt (2) durch den Datenpunkt Typ (z.B. Analog/Binary/Multistate Input/Output/Value oder Andere) bestimmt ist.

Beim *Algorithmic Reporting* ist Aspekt (1) ebenso identisch mit dem Datenpunkt aber für Aspekt (2) ist ein zusätzlicher BACnet Objekt Typ notwendig: ein Event_Enrollment Objekt.

Für Aspekt (3) ist für beide Methoden des Alarm Managements ein BACnet Objekt vom Typ Notification_Class definiert.

Grundsätzlich ist es möglich, alle Alarm Funktionalitäten **entweder** mit *Algorithmic Reporting* **oder** mit *Intrinsic Reporting* zu erfüllen.

Deshalb macht es für einen B-OWS Bediener keinen Unterschied, ob ein Gerät nur eine Art des Alarm Managements unterstützt. Mit *Intrinsic Reporting* ist es einfach eine große Anzahl von einfachen Situationen zu konfigurieren, in welchen ein Alarm generiert werden soll. Komplexe Situationen lassen sich jedoch besser mit *Algorithmic Reporting* konfigurieren, da dies wesentlich flexibler ist.

Wenn ein Gerät beide Methoden des Alarm Managements unterstützt kann die „Grundlast“ aller Alarm Situationen mit *Intrinsic Reporting* konfiguriert werden. Der Rest der komplexen Situationen kann komfortabel mit *Algorithmic Reporting* konfiguriert werden.

In diesem Dokument werden grundsätzliche Kommunikationsaspekte nicht behandelt, wie z.B.:

- bestätigte oder unbestätigte Benachrichtigungen
- Einträge als Empfänger mit den Diensten AddListElement oder WriteProperty,
- Einträge als Empfänger als BACnetObjectIdentifier or BACnetAddress,
- Alarm oder Event Benachrichtigung

1.4. Verwendete Abbildungen

Um die Konfiguration des Alarm Managements in diesem Dokument zu illustrieren sind Abbildungen verwendet. Das Design der Abbildungen ist bei anderen Herstellern wahrscheinlich verschieden, nicht jedoch das Prinzip der Konfiguration.

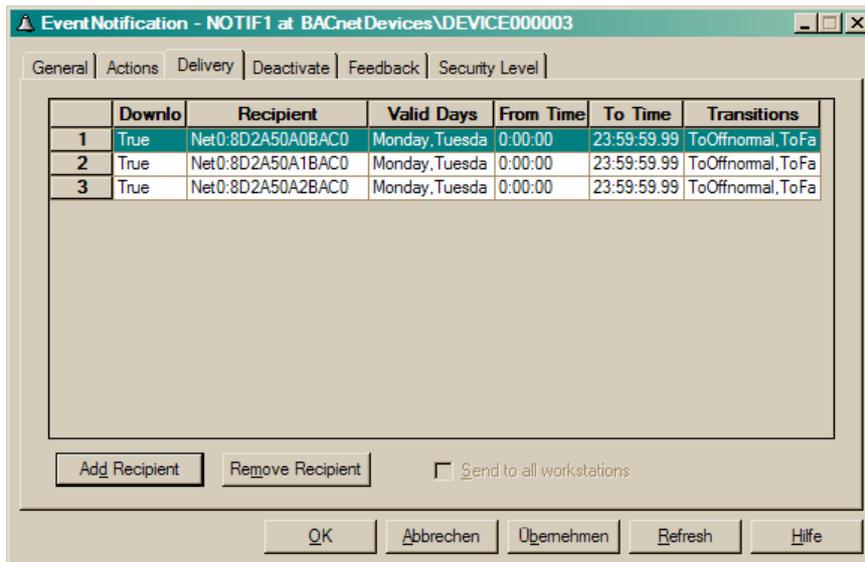
1.5. Last but not least

Vielen Dank an Norbert Schmalstieg für Kommentare und Korrektur!

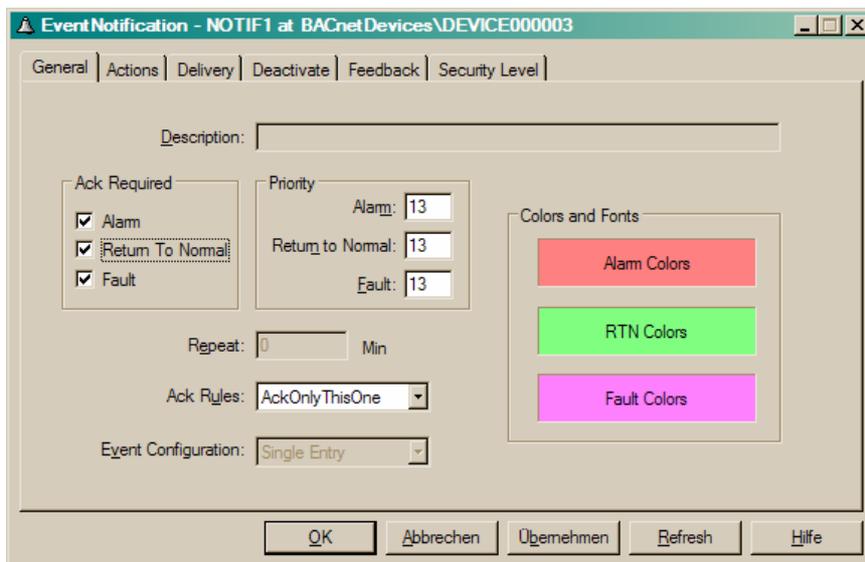
2. Beschreibung des Engineering

2.1. Allgemeines

Beide Methoden des Alarm Managements erfordern ein oder mehrere Notification_Class Objekte in einem Gerät. Neben anderen Dingen enthält dieser BACnet Objekt Typ Informationen darüber **wer** (Empfänger, B-OWS, BACnet Client...) und **wann** (an welchen Tagen und welchem Zeitfenster) als Empfänger einen ausgelösten Alarm enthält:



Weiterhin ist festgelegt, ob der Alarm eine Quittierung erfordert oder nicht:

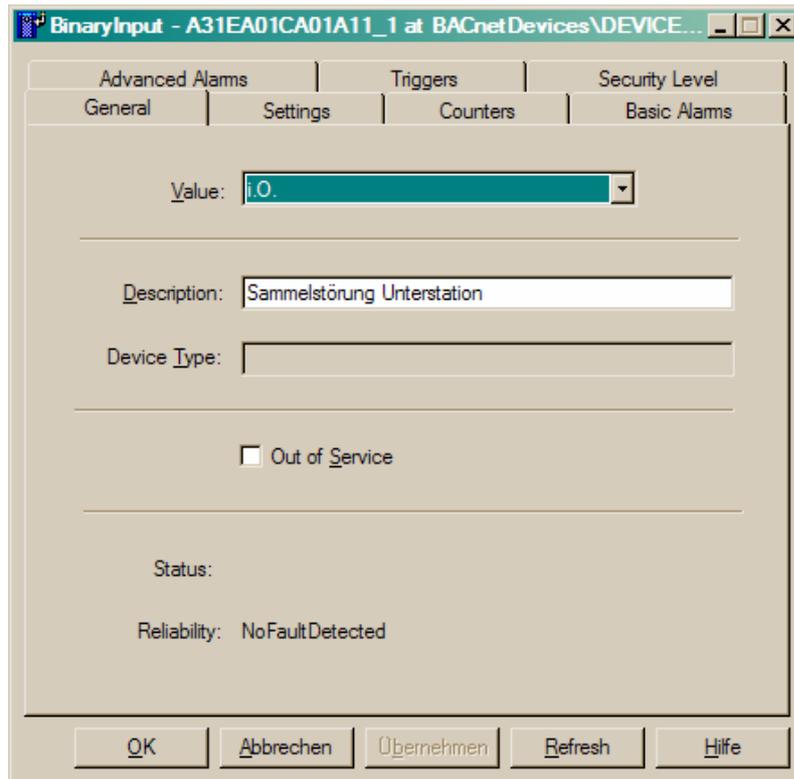


Andere Einstellungen sind spezifisch für den Hersteller und nicht im BACnet Standard definiert (Regeln für Ausdruck, Farben an der B-OWS, Mail Weiterleitung...). Solche Einstellungen sind allerdings nicht relevant, wenn beide Methoden des Alarm Managements verglichen werden.

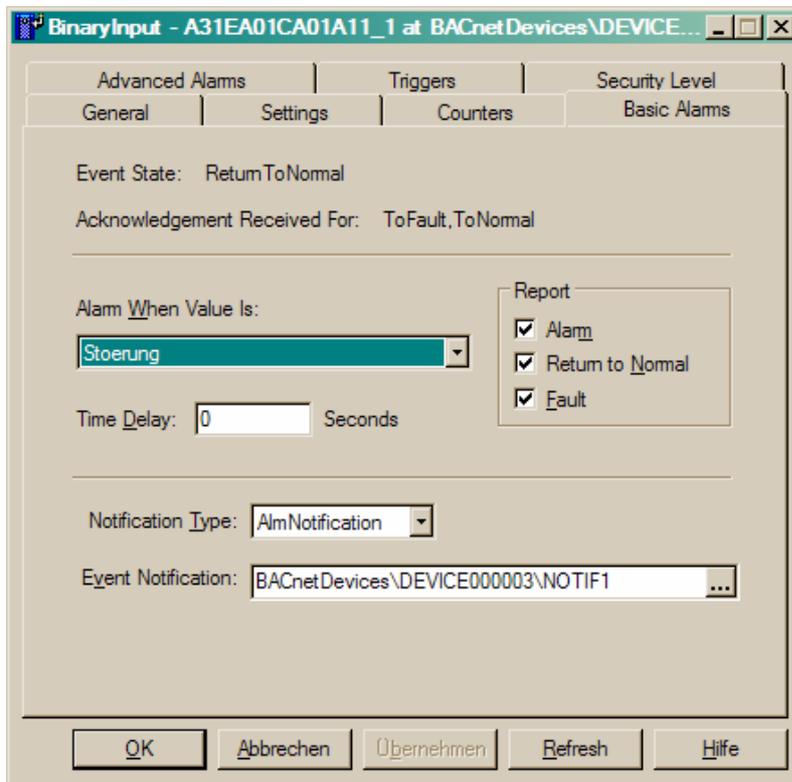
2.2. Intrinsic Reporting

Diese Methode der Ereignisbenachrichtigung ist Bestandteil eines Datenpunktes, z.B. eines Binary_Input, Analog_Value ... Das bedeutet, dass die Bedingungen für die Erzeugung eines Alarms mit eigenen Attributen des Datenpunktes definiert sind. Es besteht eine Referenz von diesem Datenpunkt zu einem Notification_Class Objekt. Kein zusätzliches BACnet Objekt ist für die Alarm Bearbeitung notwendig.

Beispiel: ein Binary_Input



In welchem Fall ein Alarm ausgelöst werden soll ist in der nächsten Abbildung dargestellt, ebenso die Referenz zu einem Notification_Class Objekt (in der Abbildung bezeichnet als Event Notification).



Wenn die Alarmbedingungen nicht ausreichend mit *Intrinsic Reporting* erfüllt werden können, ist auch eine Kombination beider Methoden der Ereignisbenachrichtigung möglich. Zur Erzeugung z.B. eines Alarms der Kategorie „Warnung“ kann basierend auf der Änderung des Present_Value eines Analog_Input Objektes zusätzlicher Alarm mit *Intrinsic Reporting* behandelt werden (siehe unten).

2.3. Algorithmic Reporting

Für diese Methode der Ereignisbenachrichtigung ist ein zusätzliches BACnet Objekt notwendig: ein Event_Enrollment Objekt.

Die Bedingungen, unter welchen ein spezieller Alarm ausgelöst werden soll, sind in einem Event_Enrollment Objekt konfiguriert.

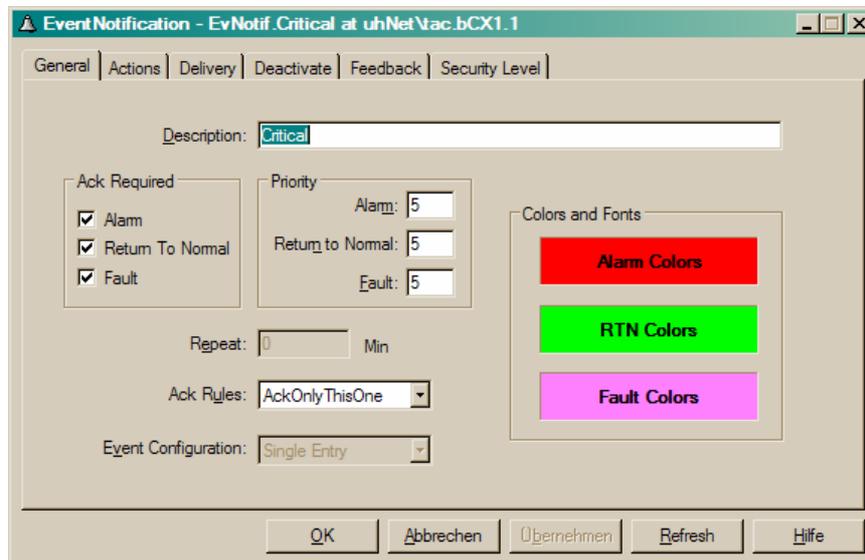
Zur Weiterleitung an Alarm Empfänger besteht eine Referenz von diesem Event_Enrollment Objekt zu einem Notification_Class Objekt.

Für vielschichtige technologische Situationen bei denen ein eine Alarmierung notwendig ist, kann für jede Situation je ein separates Event_Enrollment Objekt verwendet werden. Das macht *Algorithmic Reporting* sehr flexibel.

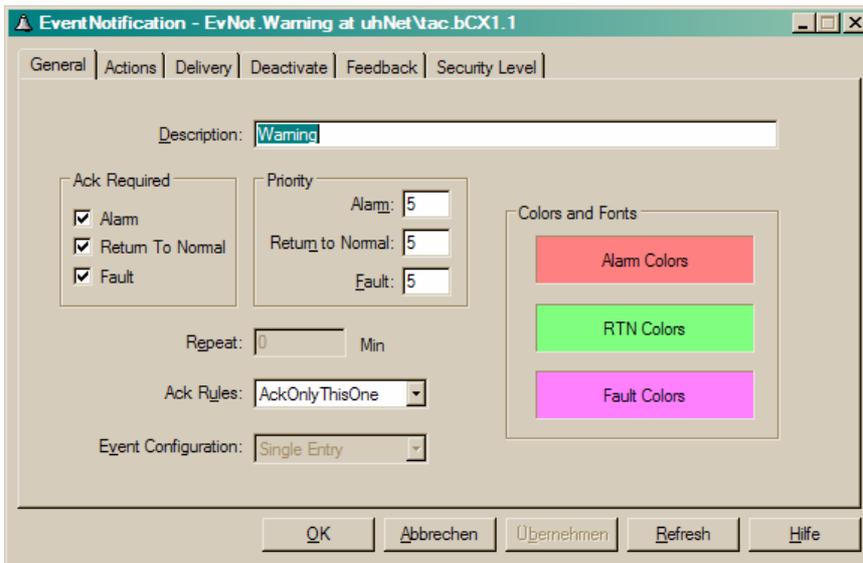
Beispiel: ein Analog_Input Objekt wird mit zwei verschiedenen Grenzwertpaaren (Warnung, Kritisch) überwacht.

Als Minimum ist ein Notification_Class Objekt erforderlich. Doch in den meisten praktischen Fällen macht es Sinn, zwei von diesen Objekten zu verwenden um einen Unterschied in der Verteilung von Warnung bzw. Kritischen Alarmen zu haben. Wenn in einem BACnet Gerät andere ähnliche Alarm Situationen für ähnliche Datenpunkte konfiguriert werden sollen, sind separate Event_Enrollment Objekte erforderlich, nicht jedoch weitere Notification_Class Objekte.

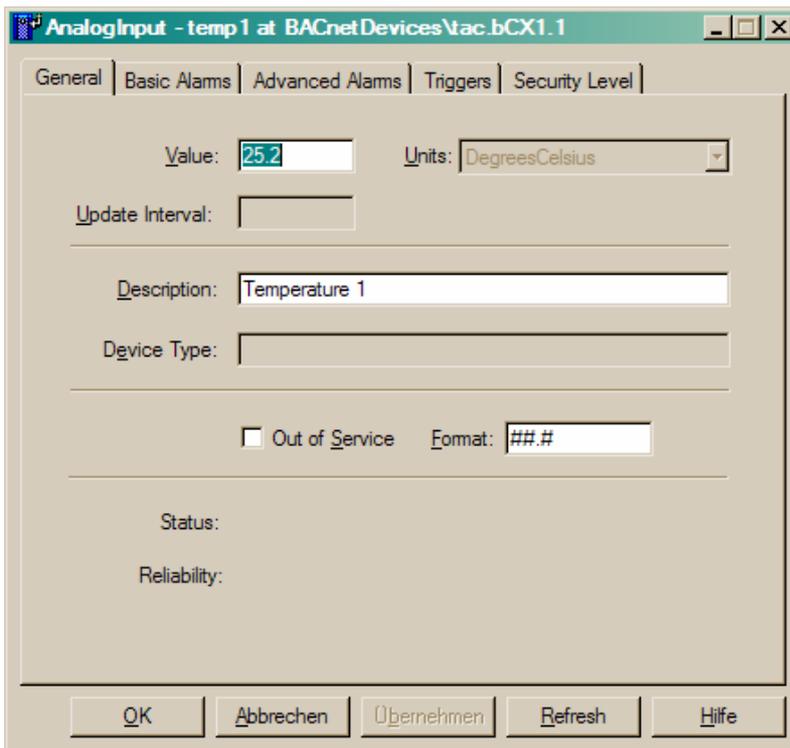
Ein Event_Enrollment Objekt für alle kritischen Alarme:



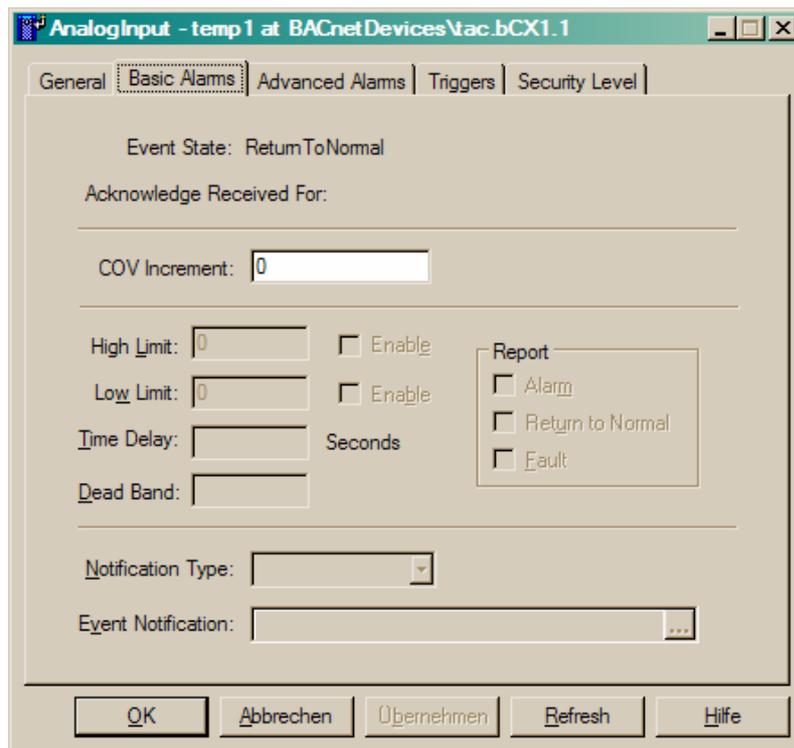
Ein zweites Event_Enrollment Objekt für alle Alarme vom Typ Warnung:



und ein Analog_Input Objekt als überwachter Datenpunkt:

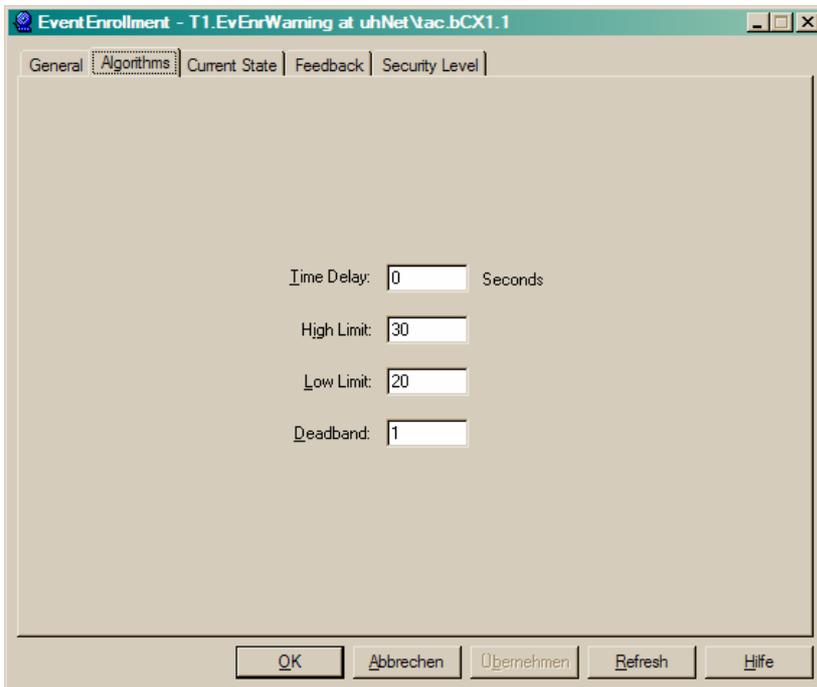
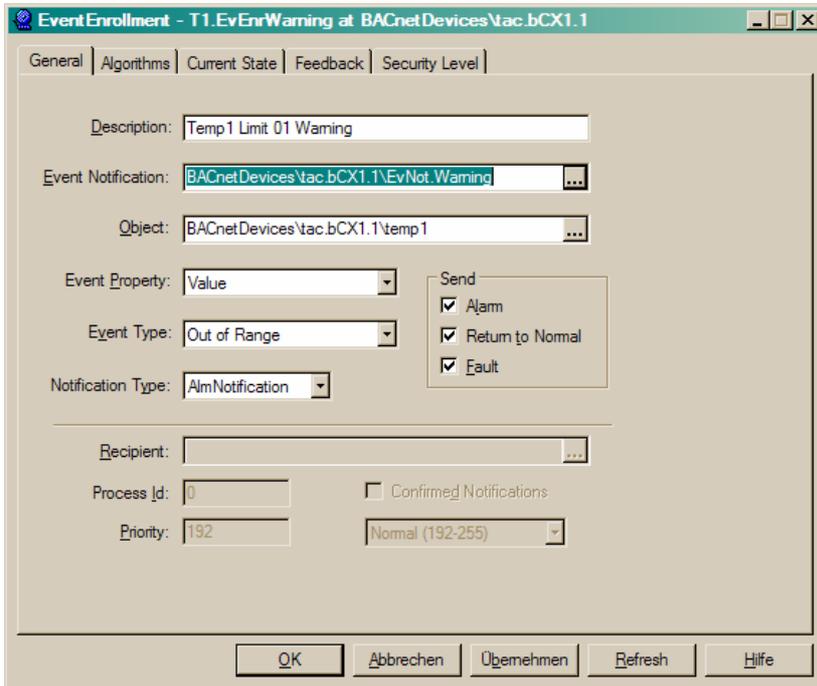


ohne ein eigenes (intrinsic) Alarm Konfigurationsattribut:



Die Beziehung zwischen einem Datenpunkt Attribut und einer Alarmauslösung ist in einem Event_Enrollment Objekt konfiguriert. In dem Beispiel sind zwei Objekte vom Typ Event_Enrollment verwendet (reines *Algorithmic Reporting*):

Das erste Event_Enrollment Objekt für Alarme vom Typ Warnung für den Datenpunkt „temp1“:



und das zweite Event_Enrollment Objekt für den kritischen Alarm des Datenpunktes „temp1“:

Event Enrollment - T1.EvEnrCritical at BACnetDevices\tac.bCX1.1

General | Algorithms | Current State | Feedback | Security Level

Description: Temp 1 Limit 01 Critical

Event Notification: BACnetDevices\tac.bCX1.1\EvNotif.Critical

Object: BACnetDevices\tac.bCX1.1\temp1

Event Property: Value

Event Type: Out of Range

Notification Type: AlmNotification

Send

- Alarm
- Return to Normal
- Fault

Recipient:

Process Id: 0

Priority: 192

Confirmed Notifications

OK Abbrechen Übernehmen Refresh Hilfe

Event Enrollment - T1.EvEnrCritical at BACnetDevices\tac.bCX1.1

General | Algorithms | Current State | Feedback | Security Level

Time Delay: 0 Seconds

High Limit: 40

Low Limit: 10

Deadband: 1

OK Abbrechen Übernehmen Refresh Hilfe

Beide Objekte überwachen denselben Datenpunkt und haben denselben Ereignis Typ (out of range). Sie unterscheiden sich in Ihrer Referenz zu dem Notification_Class Objekt und den konfigurierten Grenzwerten.

Eine andere Möglichkeit wäre, einen der Alarme mit *Intrinsic Reporting* (z.B. den Alarm vom Typ Warnung) einzurichten, den anderen mit *Algorithmic Reporting* (z.B. den kritischen Alarm).

3. Vergleich

3.1. Ingenieurtechnische Bearbeitung

<i>Intrinsic Reporting</i>	<i>Algorithmic Reporting</i>
<p>(+) Leicht zu konfigurieren und zu prüfen für einfache Alarme:</p> <ul style="list-style-type: none"> - Der Aspekt, in welcher Situation ein Alarm ausgelöst werden soll, ist im Datenpunkt enthalten. - Kein weiteres BACnet Objekt ist notwendig, um einen Alarm auszulösen. 	<p>(-) Für einfache Alarme:</p> <ul style="list-style-type: none"> - Es ist mehr Aufwand für den Projektingenieur notwendig. - Zusätzliche BACnet Objekte sind notwendig.
<p>(-) Es ist nicht möglich, direkt komplexe Alarme zu konfigurieren z.B.:</p> <ul style="list-style-type: none"> - Wenn eine Temperatur mit mehreren Grenzwertpaaren überwacht werden soll (Warnung, kritisch...). - Wenn mehr als ein Attribut (z.B. Present_Value, Out_Of_Service) alarmauslösend sein soll. - Wenn gleitende Grenzwerte den Alarm auslösen sollen (nur ein unveränderliche Grenzen sind definiert, vergleichbar mit dem Ereignis Typ Out_Of_Range). <p>Für eine kleine Anzahl von Objekt Typen ist nicht unterstützt. Zum Beispiel diese Objekte:</p> <ul style="list-style-type: none"> - Device Object Type - Program Object Type - Schedule Object Type <p>Um dennoch <i>Intrinsic Reporting</i> zu verwenden sind proprietäre Erweiterungen notwendig.</p>	<p>(+) Sehr flexibel um komplexe Alarme zu konfigurieren:</p> <ul style="list-style-type: none"> - Durch den BACnet Standard ist eine große Anzahl von verschiedenen Event Typen vordefiniert: {CHANGE_OF_BITSTRING, CHANGE_OF_STATE, CHANGE_OF_VALUE, COMMAND_FAILURE, FLOATING_LIMIT, OUT_OF_RANGE, BUFFER_READY, CHANGE_OF_LIFE_SAFETY, EXTENDED} - Alle erforderlichen Alarm Funktionen können direkt mit einer hohen Anzahl von Notification_Class Objekten realisiert werden.

<i>Intrinsic Reporting</i>	<i>Algorithmic Reporting</i>
<p>(-) Wenn <i>Algorithmic Reporting</i> nicht unterstützt wird, können z.B. mit gespiegelten Datenpunkten¹ und proprietärer Programmierung komplexe Alarmer behandelt werden. Ein derartiges Engineering ist allerdings nicht transparent aus der Sicht einer BACnet Bedienstation.</p> <p>Wenn <i>Algorithmic Reporting</i> unterstützt wird, Elemente von <i>Algorithmic Reporting</i> können hinzugefügt werden, was dann zu einer Kombination beider Methoden des Alarm Managements führt.</p>	<p>(+) Alle Alarm Konfigurationen sind transparent weil sie exklusiv in den Event_Enrollment Objekten konfiguriert sind.</p>
<p>(+) Nach einem Alarm Ereignis ist es einfach zu erkennen, welcher Datenpunkt dieses Ereignis ausgelöst hat, da es sich genau um denselben Datenpunkt handelt.</p>	<p>(-) Nach einem Alarm Ereignis benötigt es mehr Aufwand um zu erkennen, welcher Datenpunkt dieses Ereignis ausgelöst hat, weil das bezügliche Event_Enrollment Objekt lediglich eine Referenz zu dem verantwortlichen Datenpunkt hat.</p>

Falls ein Hersteller beide Methoden des Alarm Managements unterstützt, können vom Projektgenieur einfache Alarmer mit *Intrinsic Reporting* und komplexe Alarmer mit *Algorithmic Reporting* definiert werden.

Falls ein Hersteller nur eine der beiden Methoden des Alarm Managements unterstützt, ist die Definition von Alarmen allerdings auch für alle technologisch notwendigen Situationen möglich.

Falls ein Hersteller nur *Intrinsic Reporting* unterstützt, können zusätzliche gespiegelte Datenpunkte erzeugt werden, bei denen das Attribut Present_Value denselben Wert hat. Jeder originale und gespiegelte Datenpunkt kann dann auf unterschiedliche Art mit *Intrinsic Reporting* definiert werden.

Allerdings wird durch *Intrinsic Reporting* nur eine geringere Anzahl von Attributen, Objekt Typen und Event Typen unterstützt als durch *Algorithmic Reporting*, so dass in solchem Fall proprietäre Erweiterungen notwendig sein können.

¹ Zusätzlicher Datenpunkt (z.B. Analog_Value Objekt), bei welchem der Present_Value durch eine proprietäre Programmierung auf den Present_Value eines anderen Datenpunktes (z.B. Analog_Input Objekt), welches einen physikalischen Eingang darstellt.

3.2. BACnet Oberfläche für Benutzer / Bediener

Mit beiden Methoden der Ereignisbenachrichtigung können alle technologischen Situationen in welchen ein Alarm notwendig ist beherrscht werden. Komplexe Situationen können allerdings nicht ausschließlich mit BACnet Engineering behandelt werden, proprietäre Ergänzungen müssen möglich sein (siehe oben).

Für eine B-OWS oder einen anderen Client ist es dasselbe, ob eine Meldung über *Intrinsic Reporting* oder *Algorithmic Reporting* gesendet wurde. Ein Client muss in der Lage sein, Meldungen von beiden Methoden der Ereignisbenachrichtigung zu empfangen.

Die Behandlung von komplexen Alarmen mit *Intrinsic Reporting* kann ein Nachteil für den Benutzer der BACnet Bedienoberfläche sein, weil es nicht möglich ist die Beziehungen zwischen den originalen und den gespiegelten Datenpunkten zu erkennen. Im Fall des *Algorithmic Reporting* ist diese Beziehung transparent konfiguriert im Event_Enrollment Objekt.

Letztendlich ist der Aufwand zur Erstellung eines BACnet Projektes nicht wirklich interessant für den Benutzer. Dieser Aufwand hängt auch von den Software Werkzeugen des Herstellers ab.

3.3. Zusammenfassung

Es gibt keinen Grund *Algorithmic Reporting* oder *Intrinsic Reporting* in Ausschreibungen auszuschließen. Jedoch ist *Algorithmic Reporting* mehr flexibel als *Intrinsic Reporting*.

In einer Ausschreibung ist es wesentlich mehr wichtig die technologischen Aspekte des Alarmmanagements zu beschreiben als eine bestimmte Methode der Ereignisbenachrichtigung (*Intrinsic Reporting* oder *Algorithmic Reporting*) zu fordern.

4. Andere Veröffentlichungen im Zusammenhang mit beiden Methoden der Ereignisbenachrichtigung

In [1] (Kapitel 13) sind Dienste für Ereignisbenachrichtigung definiert. In [1] wird weder *Intrinsic Reporting* noch *Algorithmic Reporting* bevorzugt.

In [2] (Kapitel 9) werden zusätzliche Regeln für die Implementierung von Diensten für die Ereignisbenachrichtigung definiert. In [2] wird weder *Intrinsic Reporting* noch *Algorithmic Reporting* bevorzugt.

In [3] (Kapitel 5.2) finden wir eine Erläuterung der Dienste für die Ereignisbenachrichtigung. In [3] wird weder *Intrinsic Reporting* noch *Algorithmic Reporting* bevorzugt.

In [4] (Kapitel 3.3.1 Seite 18) wird *Intrinsic Reporting* ohne weitere Begründung bevorzugt. Unter Berücksichtigung der Beispiele und Fakten in diesem Beitrag kann ich dieser Ansicht nicht folgen.

In [5] (Kapitel 5) finden wir eine umfassende Erläuterung der Ereignisbenachrichtigung. In [2] wird weder *Intrinsic Reporting* noch *Algorithmic Reporting* bevorzugt.

5. Literatur

- [1] ANSI/ASHRAE Standard 135-2004 (BACnet Standard)
- [2] BTL Implementers Guide-v16
- [3] BACnet_Implementation_Guide_V2.pdf (BACnet Handbuch der BIG-EU)
- [4] VDI-GA-BIG-EU-BACnet-Leitfaden V2_5-05-07-22
- [5] BACnet Gebäudeautomation 1.4, Hans R. Kranz