Bedienungsanleitung für Analytics Builder for Cumulocity IoT

Version 10.15.0

November 2022

APAMA

This document applies to Analytics Builder for Cumulocity IoT 10.15.0 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2018-2022 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at https://softwareag.com/licenses/.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at https://softwareag.com/licenses/ and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at https://softwareag.com/licenses/ and/or in the root installation directory of the licensed product(s).

Document ID: PAB-UG-10150-20221114_DE

Inhaltsverzeichnis

Über diese Dokumentation....7
Dokumentations-Roadmap....8
Dokumentationskonventionen....8
Online-Informationen und Support....9
Datenschutz....10

1 Release Notes....11
Was ist neu in Version 10.15.0....12
Was ist neu in Version 10.14.0....12
Was ist neu in Version 10.13.0....13
Was ist neu in Version 10.11.0....13
Was ist neu in Version 10.10.0....14
Was ist neu in Version 10.9.0....14
Was ist neu in Version 10.7.0....17
Was ist neu in Version 10.6.6....19
Was ist neu in Version 10.6....19
Was ist neu in Version 10.5.7....19
Was ist neu in Version 10.5....20
Was ist neu in Version 10.3.2....22
Was ist neu in Version 10.3.1....23

2 Einführung in Analytics Builder....25
Was ist Analytics Builder?....26
Analytics Builder und Cumulocity IoT....26
Voraussetzungen....28
Analytics Builder starten....28
Spracheinstellungen....29
Erste Schritte: Erstellen Sie Ihr erstes Modell....29
Erste Schritte: Erstellen Sie ein Modell aus einem Beispiel....33

3 Modelle verstehen....39
Modelle....40
Vorlagenmodellinstanzen....41
Blöcke....41
Verbindungen....45
Anwendungsbeispiel....45

4 Mit dem Modellmanager arbeiten....49
Die Benutzeroberfläche des Modellmanagers....50
Modelle und Beispiele filtern....52
Ein neues Modell erstellen....54
Ein bestehendes Modell bearbeiten....55
Die Instanzen eines Modells bearbeiten....56

Ein Modell aktivieren....56
Ein Modell deaktivieren....58
Ein Modell duplizieren....58
Ein Modell exportieren....59
Ein Modell importieren....59
Ein Modell entfernen....59
Alle Modelle neu laden....60
Ein Beispiel anzeigen....60
Ein Modell aus einem Beispiel erstellen....60

5 Mit dem Modelleditor arbeiten....63
Die Benutzeroberfläche des Modelleditors....64
Mit Modellen arbeiten....65
Mit Blöcken und Verbindungen arbeiten....67
Mit Gruppen arbeiten....81
Die Arbeitsfläche verwalten....87

6 Mit dem Instanzeditor arbeiten....91
Die Benutzeroberfläche des Instanzeditors....92
Eine Instanz erstellen....93
Eine Instanz editieren....93
Eine Instanz aktivieren....94
Eine Instanz deaktivieren....95
Die Instanzen filtern und sortieren....95
Eine Instanz duplizieren....96
Eine Instanz entfernen....96
Die Instanzen speichern....96
Die Instanzen neu laden....97
Den Instanzeditor verlassen....97

7 Verbindungen und Blöcke....99
Über eine Verbindung gesendete Werte....100
Typumwandlungen....106
Verarbeitungsreihenfolge der Verbindungen....108
Verbindungseinschränkungen....109
Ein- und Ausgänge der Blöcke....110
Häufig vorkommende Blockeingänge und Parameter....110
Eingabeblöcke und Ereignissteuerung....112
Ausgabeblöcke und Ereignissteuerung....113
Fragmenteigenschaften auf Verbindungen....114
Schlüssel zur Identifizierung einer Ereignisreihe....114

8 Details zu Werten und Blöcken....117
Einleitung....118
Werte als Repräsentationen von zeitkontinuierlichen physikalischen Größen....118
Zeitfenster für die Ausgabesteuerung eines Blocks....125
Pulse-Signale....128
Messungen von diskreten Zeitwerten....130

9 Modelle und Geräte....133
Modellausführung für verschiedene Geräte....134
Broadcast-Geräte....136
Virtuelle Geräte....137
Verbindungen zwischen Modellen....138
Die Anzahl der angezeigten Geräte, Gerätegruppen und/oder Assets konfigurieren....141
Nach Eingabe- und Ausgabe-Assets suchen....141

10 Modellsimulation....143
Allgemeine Informationen zum Simulationsmodus....144
Simulationsparameter....145
Die maximale Anzahl von Simulationsmodellen konfigurieren....145
Eine alternative Datenquelle für die Simulation konfigurieren....145
Verworfene Eingaben überwachen....147

11 Überwachung und Konfiguration....149
Überwachung....150
Konfiguration....157
Korrelatorlog anzeigen....161

12 Blockreferenz....163
Übersicht aller Blöcke....164
Eingabe....167
Ausgabe....176
Logik....184
Berechnung....186
Aggregat....203
Fluss-Manipulation....212
Werkzeug....221

Über diese Dokumentation

- Dokumentations-Roadmap 8
- Dokumentationskonventionen 8
- Online-Informationen und Support 9
- Datenschutz 10

Diese Dokumentation beschreibt wie Sie Modelle mit Analytics Builder for Cumulocity IoT erstellen und benutzen.

Dokumentations-Roadmap

Diese Dokumentation steht in den folgenden Formaten zur Verfügung:

- HTML
- PDF

Zusätzlich stehen im Modelleditor (dies ist eines der Tools von Analytics Builder for Cumulocity IoT) Referenzinformationen zu den vorgefertigten Blöcken zur Verfügung (Blockbeschreibungen, Parameter, sowie Details zu den Ein- und Ausgangsports).

Dokumentationskonventionen

Konvention	Beschreibung
Fettschrift	Kennzeichnet Elemente auf einem Bildschirm.
Schmal geschnittene Schrift	Kennzeichnet Namen und Orte von Diensten im Format *Ordner.Unterordner.Dienst*, Programmierschnittstellen (APIs), Namen von Klassen, Methoden und Properties in Java.
Kursivschrift	Kennzeichnet: Variablen, für die Sie situations- oder umgebungsspezifische Werte angeben müssen. Neue Begriffe, wenn sie erstmals im Text auftreten. Verweise auf andere Dokumentationsquellen.
`Nichtproportionale Schrift`	Kennzeichnet: Text, den Sie eingeben müssen. Meldungen, die vom System angezeigt werden. Programmcode.
{ }	Zeigt eine Reihe von Auswahlmöglichkeiten an, von denen Sie eine auswählen müssen. Geben Sie nur die innerhalb der geschweiften Klammern vorhandenen Informationen ein. Geben Sie nicht die Klammersymbole { } ein.
|	Trennt zwei sich gegenseitig ausschließende Auswahlmöglichkeiten in einer Syntaxzeile voneinander ab. Geben Sie eine der Auswahlmöglichkeiten ein. Geben Sie nicht das Symbol | ein.
[ ]	Zeigt eine oder mehrere Optionen an. Geben Sie nur die innerhalb der eckigen Klammern vorhandenen Informationen ein. Geben Sie nicht die Klammersymbole [ ] ein.

Konvention	Beschreibung
...	Zeigt an, dass Sie mehrere Auswahlmöglichkeiten desselben Typs eingeben können. Geben Sie nur die Informationen ein. Geben Sie nicht die drei Auslassungspunkte (...) ein.

Online-Informationen und Support

Produktdokumentation

Sie finden die Produktdokumentation auf unserer Dokumentationswebsite unter https://documentation.softwareag.com.

Zusätzlich können Sie auch über https://www.softwareag.cloud auf die Dokumentation für die Cloud-Produkte zugreifen. Navigieren Sie zum gewünschten Produkt und gehen Sie dann, je nach Produkt, zu „Developer Center“, „User Center“ oder „Documentation“.

Produktschulungen

Sie finden hilfreiches Produktschulungsmaterial auf unserem Lernportal unter https://knowledge.softwareag.com.

Tech Community

Auf der Website unserer Tech Community unter https://techcommunity.softwareag.com können Sie mit Experten der Software AG zusammenarbeiten. Von hier aus können Sie zum Beispiel:

- Unsere umfangreiche Wissensdatenbank durchsuchen.
- In unseren Diskussionsforen Fragen stellen und Antworten finden.
- Die neuesten Nachrichten und Ankündigungen der Software AG lesen.
- Unsere Communities erkunden.
- Unsere öffentlichen Repositories auf GitHub and Docker unter https://github.com/softwareag und https://hub.docker.com/publishers/softwareag besuchen und weitere Ressourcen der Software AG entdecken.

Produktsupport

Support für die Produkte der Software AG steht lizenzierten Kunden über unser Empower-Portal unter https://empower.softwareag.com zur Verfügung. Für viele Dienstleistungen auf diesem Portal benötigen Sie ein Konto. Wenn Sie noch keines haben, dann können Sie es unter https://empower.softwareag.com/register beantragen. Sobald Sie ein Konto haben, können Sie zum Beispiel:

- Produkte, Aktualisierungen und Programmkorrekturen herunterladen.

- Das Knowledge Center nach technischen Informationen und Tipps durchsuchen.
- Frühwarnungen und kritische Alarme abonnieren.
- Supportfälle öffnen und aktualisieren.
- Anfragen für neue Produktmerkmale einreichen.

Datenschutz

Die Produkte der Software AG stellen Funktionen zur Verarbeitung von personenbezogenen Daten gemäß der Datenschutz-Grundverordnung (DSGVO) der Europäischen Union zur Verfügung. Gegebenenfalls sind in der betreffenden Systemverwaltungsdokumentation entsprechende Schritte dokumentiert.

1 Release Notes

- Was ist neu in Version 10.15.0 12
- Was ist neu in Version 10.14.0 12
- Was ist neu in Version 10.13.0 13
- Was ist neu in Version 10.11.0 13
- Was ist neu in Version 10.10.0 14
- Was ist neu in Version 10.9.0 14
- Was ist neu in Version 10.7.0 17
- Was ist neu in Version 10.6.6 19
- Was ist neu in Version 10.6 19
- Was ist neu in Version 10.5.7 19
- Was ist neu in Version 10.5 20
- Was ist neu in Version 10.3.2 22
- Was ist neu in Version 10.3.1 23

Was ist neu in Version 10.15.0

- Im Modellmanager werden jetzt zusätzliche Analytics Builder-Beispiele zur Verfügung gestellt. Die bereitgestellten Beispiele decken nun alle Smart Rules von Cumulocity IoT ab.
- Die Analytics Builder-Beispiele haben jetzt ausführliche Beschreibungen. Sie können die ersten Zeilen einer solchen Beschreibung in einem Tooltip sehen, wenn Sie die Maus über die Karte für ein Beispiel bewegen, oder Sie können die vollständige Beschreibung sehen, wenn Sie ein Beispiel anzeigen (oder ein Modell aus dem Beispiel erstellen) und dann das Dialogfeld "Modellkonfiguration" öffnen.
- Es ist jetzt möglich, nach Analytics Builder-Beispielen mit bestimmten Wörtern in ihren Namen zu suchen und sie nach Tags zu filtern.
- In der vorherigen Version waren die Vorlagenparameternamen in den Analytics Builder-Beispielen nur in englischer Sprache verfügbar. Ab dieser Version sind Übersetzungen für die unterstützten Sprachen verfügbar.
- Die Blöcke Alarmausgabe und Messwertausgabe verfügen jetzt über einen neuen **Parameter-Fragment**-Parameter und einen neuen **Eigenschaften**-Eingangsport. Diese können dazu verwendet werden, benutzerdefinierte Eigenschaften festzulegen, ähnlich der bereits vorhandenen Funktionalität des Ereignisausgabe-Blocks.
- Der Alarmausgabe-Block hat jetzt einen neuen **Asynchrone Ausgabe erstellen**-Parameter. Er ist standardmäßig nicht aktiviert, was das vorherige Verhalten des Block widerspiegelt. Wenn dieser Parameter aktiviert ist, können Alarme nun asynchron erzeugt und gelöscht werden.

  ACHTUNG:
  Wenn Sie den **Asynchrone Ausgabe erstellen**-Parameter aktivieren ist besondere Vorsicht geboten. Dies kann zu Endlosschleifen während der Modellausführung führen (für dasselbe Modell oder über mehrere Modelle hinweg), was zu Leistungseinbußen oder zu Speicherplatzmangel in der Anwendung führt.
- Analytics Builder unterstützt jetzt die Branding-Option `globalTitle` für die Kategorienamen im Modelleditor. Siehe auch Branding and language customization in der Cumulocity IoT-Dokumentation.
- Wie in „Was ist neu in Version 10.14.0“ auf Seite 12 angekündigt, wurde die Eigenschaft `frgment` in 10.15 entfernt. Falls Sie dies noch nicht getan haben, müssen Sie jetzt in allen Ihren Blöcken `frgment` in `fragment` abändern (zum Beispiel den Wert des **Eigenschaftspfad**-Parameters im Eigenschaft extrahieren-Block).

Was ist neu in Version 10.14.0

- Im Modellmanager werden jetzt Beispiele zur Verfügung gestellt. Diese ähneln den Smart Rules in Cumulocity IoT. Die Beispiele sollen Ihnen den Einstieg in die Erstellung Ihrer eigenen Modelle erleichtern. Sie können ein Beispiel als Grundlage für die weitere Entwicklung verwenden, indem Sie aus dem Beispiel ein neues Modell erstellen. Anschließend können Sie das neue Modell Ihren Anforderungen entsprechend ändern und aktivieren. Siehe auch „Erste

Schritte: Erstellen Sie ein Modell aus einem Beispiel“ auf Seite 33 und die Beschreibung der neuen Registerkarte **Beispiele** in „Die Benutzeroberfläche des Modellmanagers“ auf Seite 50.

- Der Messwertausgabe-Block hat jetzt einen neuen **Zeit**-Eingangsport. Dieser kann dazu verwendet werden, den Zeitstempel der Messung zu setzen und funktioniert ähnlich wie die bereits vorhandenen **Zeit**-Eingangsports bei den Alarmausgabe- und Ereignisausgabe-Blöcken.
- Der Messwerteingabe-Block sendet den Eigenschaftsnamen für ein Fragment als Teil seines Ausgabewerts. In früheren Versionen lautete dieser Eigenschaftsname `frgment`. Dieser Name wurde nun in `fragment` abgeändert. Zur Zeit werden beide Eigenschaftsnamen `frgment` und `fragment` unterstützt. `frgment` ist jedoch veraltet und wird in einem späteren Release entfernt. Daher wird empfohlen, dass Sie jetzt in allen Ihren Blöcken `frgment` in `fragment` abändern (zum Beispiel den Wert des **Eigenschaftspfad**-Parameters im Eigenschaft extrahieren-Block).
- Der Messwerteingabe-Block sendet jetzt auch eine neue Eigenschaft mit dem Namen `value` als Teil seines Ausgabewerts. Dadurch können andere Blöcke, wie zum Beispiel der Textersetzung-Block, den Messwert verwenden (zum Beispiel, in einem Alarmtext).

Bekannte Probleme

- Die Vorlagenparameternamen in den Beispielen sind derzeit nur in englischer Sprache verfügbar. Es ist geplant, sie in einer späteren Version in die unterstützten Sprachen zu übersetzen.

Was ist neu in Version 10.13.0

- Die Blöcke E-Mail senden und SMS senden können jetzt im Simulations- und Testmodus verwendet werden. Ein E-Mail oder SMS wird in diesen Fällen jedoch nicht gesendet. Statt dessen wird die Ausgabe dieser Blöcke protokolliert.

  In vorherigen Versionen schlug die Modellaktivierung im Simulations- oder Testmodus bei der Verwendung dieser Blöcke mit der folgenden Fehlermeldung fehl: `Model produces output for device '' not declared by any block. Make sure that device parameters of all blocks are properly tagged.`
- Der Textersetzung-Block unterstützt jetzt den neuen `FORMAT`-Parameter zur Angabe eines anderen Zeitformats. So können Sie zum Beispiel `#{time:TZ="America/New_York",FORMAT="HH:mm:ssZ"}` angeben, um die Zeitzone für New York zu verwenden und das Zeitformat auf `HH:mm:ssZ` zu setzen. Eine Liste der gültigen Zeichenfolgen für das Zeitformat finden Sie in der Apama-Dokumentation unter Format specification for the TimeFormat functions.

Was ist neu in Version 10.11.0

Diese Version von Analytics Builder enthält keine neue Funktionalität.

Was ist neu in Version 10.10.0

- Ein Problem wurde behoben, welches Verbindungen verhinderte zwischen Blöcken mit Ausgaben vom Typ `any` und Blöcken, die `float` als Eingabe akzeptieren. Außerdem wurde der Ausdruck-Block aktualisiert, damit Eingaben vom Typ `any` verarbeitet werden können.
- Der Pulse-Block hat jetzt einen neuen **Modus**-Parameter. Die Standardeinstellung ist identisch mit dem bisherigen Verhalten des Blocks: der Block sendet ein Pulse-Signal, wenn sich der Wert ändert. Mit diesem neuen Parameter können Sie den Block jetzt auch so konfigurieren, dass ein Pulse-Signal bei jeder Eingabe oder bei Nicht-Null-Werten gesendet wird. Siehe auch „ Typumwandlungen“ auf Seite 106.
- Der Textersetzung-Block unterstützt jetzt Zeitzonenwerte. Sie können jetzt ein Suffix `:TZ=zeitzone` angeben, um eine andere Zeitzone zu verwenden, zum Beispiel, `#{time:TZ=America/New_York}` um die Zeitzone für New York zu verwenden.
- Der **Eigenschaftsname**-Parameter des Eigenschaft extrahieren-Blocks wurde in **Eigenschaftspfad** umbenannt, da dieser Parameter auch einen Pfad akzeptiert. Dies ist nur eine Änderung der Bezeichnung. Vorhandene Modelle müssen nicht geändert werden; sie funktionieren weiterhin wie bisher.

Was ist neu in Version 10.9.0

- Im Application Switcher von Cumulocity IoT steht jetzt eine neue Anwendung mit dem Namen „Streaming Analytics“ zur Verfügung. Analytics Builder ist jetzt ein Bestandteil dieser neuen Anwendung, zusammen mit EPL Apps. Siehe auch „Analytics Builder starten“ auf Seite 28.
- Die folgenden neuen Blöcke stehen jetzt zur Verfügung:
  - Verhalten von Smart Rules::
    - KPI (dieser Block implementiert den Kernbestandteil der Smart Rule "Bei Schwellwert Alarm erzeugen"; siehe auch Smart Rules-Sammlung im deutschsprachigen Cumulocity IoT *User guide*)
    - Textersetzung (siehe auch Smart Rule-Variablen im deutschsprachigen Cumulocity IoT *User guide*)
  - Unterstützung für Position und Geofence:
    - Positionseingabe (neuer Blocktyp bei den Eingabeblöcken)
    - Geofence
  - Ein- und Ausgaben für Nicht-Geräte (siehe auch die Smart Rules "Bei Alarm SMS senden" und "Bei Alarm E-Mail senden" unter Smart Rules-Sammlung im deutschsprachigen Cumulocity IoT *User guide*):
    - Konstanter Wert (siehe auch die unten stehende Liste der neuen Ein- und Ausgangsports bei den bestehenden Blöcken)
    - Cron Timer

        - SMS senden
        - E-Mail senden
    - Weitere Berechnungen und Aggregate:
        - Von Basis N
        - Nach Basis N
        - Limit
        - Bereich
        - Diskrete Statistik (siehe auch „Details zu Werten und Blöcken“ auf Seite 117)
        - Gruppenstatistik
    - Werkzeuge und Fluß-Manipulation für mehr Flexibilität beim Schreiben von Modellen:
        - Dauer
        - Eigenschaften festlegen (siehe auch die unten stehende Liste der neuen Ein- und Ausgangsports bei den bestehenden Blöcken)
        - Selektor
        - Schalter (dieser Block ist für die Verwendung mit Vorlagenparametern gedacht)

    Detaillierte Informationen zu den neuen Blöcken finden Sie in der „Blockreferenz“ auf Seite 163.
- Erweiterungen und Änderungen an bestehenden Blöcken:
    - Managed-Object-Eingabe:
        - Der Parameter **Eigenschaftsname** ist jetzt optional. Wenn er nicht gesetzt ist, dann generiert jede Aktualisierung eines Managed Object eine Ausgabe.
    - Managed-Object-Ausgabe:
        - Der Parameter **Eigenschaftsname** ist jetzt optional, so dass mehrere Eigenschaften auf einen Schlag aktualisiert werden können (zum Beispiel mit dem neuen **Eigenschaften festlegen**-Block).
    - Ereignisausgabe:
        - Neuer **Zeit**-Eingangsport.
        - Neuer **Eigenschaften**-Eingangsport (siehe auch den neuen Block **Eigenschaften festlegen**).
    - Operationsausgabe:
        - Neuer **Eigenschaften**-Eingangsport (siehe auch den neuen Block **Eigenschaften festlegen**).
    - Alarmausgabe:

- Der **Schweregrad**-Parameter ist jetzt optional. Der Schweregrad eines Alarms kann jetzt entweder mit dem **Schweregrad**-Parameter oder dem neuen **Schweregrad**-Eingangsport gesetzt werden. Eins von beiden, Eingangsport oder Parameter, muss verbunden oder gesetzt sein.
- Der **Schweregrad**-Eingangsport (aber nicht der Parameter) unterstützt jetzt auch den Wert "GELÖSCHT" damit ein Alarm gelöscht werden kann (anstatt den Schweregrad eines vorhandenen Alarms zu erstellen oder zu ändern).
- Der **Nachricht**-Parameter ist jetzt optional. Die Nachricht eines Alarms kann jetzt entweder mit dem **Nachricht**-Parameter oder dem neuen **Nachricht**-Eingangsport gesetzt werden. Eins von beiden, Eingangsport oder Parameter, muss verbunden oder gesetzt sein.
- Neuer **Zeit**-Eingangsport.
- Neuer **Alarm löschen**-Eingangsport.

- Eigenschaft extrahieren:
  - Der **Eigenschaftstyp**-Parameter unterstützt nun eine neue Option: **Eigenschaften**. Diese ist für die Verwendung mit den folgenden Blocktypen gedacht: **Operationsausgabe** (der **Eigenschaften**-Eingangsport), **Ereignisausgabe** (der **Eigenschaften**-Eingangsport), und **Managed-Object-Ausgabe** (der **Wert**-Eingangsport).
  - Der **Eigenschaftsname**-Parameter ist jetzt optional so dass der **Eigenschaftstyp** die Eigenschaften einfach durchreichen kann (siehe auch den neuen **Eigenschaften festlegen**-Block).

  Detaillierte Informationen zu den oben genannten Änderungen finden Sie in der „Blockreferenz“ auf Seite 163.

- Die Eingabeblöcke können nun zwischen Erstellungs- und Aktualisierungsereignissen unterscheiden. Zu diesem Zweck gibt es im Blockparametereditor jetzt den neuen Parameter **Benachrichtigungsmodus**. Dieser Parameter ist zum Beispiel dann hilfreich, wenn Sie sich nur für neu erstellte Alarme interessieren und nicht für Modelle, die aktiviert werden, wenn derselbe Alarm aktualisiert wird. Dieser neue Parameter ist verfügbar für die Eingabeblocktypen **Alarmeingabe**, **Ereigniseingabe**, **Operationseingabe** und auch für den neuen Typ **Positionseingabe**. Er ist nicht verfügbar für den Eingabeblocktyp **Messwerteingabe**, da Messwerte nicht aktualisiert werden können. Weitere Informationen finden Sie in der Blockreferenz in der Kategorie „Eingabe“ auf Seite 167.

- Der neue Parameter `chain_diagnostics` steht jetzt für die Überwachung des regelmäßig wiederkehrenden Status zur Verfügung. Er bietet Diagnoseinformationen zu Modellketten, wie zum Beispiel den Erstellungszeitpunkt der Kette oder wie oft die Kette ausgewertet wurde. Weitere Informationen finden Sie unter „Regelmäßig wiederkehrenden Status überwachen“ auf Seite 150.

- Der neue Key `block_promise_timeout_secs` steht jetzt für die Konfiguration der Modell-Timeouts zur Verfügung. Diese Mandantenoption definiert das Zeitlimit in Sekunden, um auf die Erledigung von `Promise`-Werten zu warten, die von Blockaktionen wie `$validate` zurückgegeben

werden. Der Standardwert ist 60 Sekunden. Weitere Informationen finden Sie unter „Keys für Modell-Timeouts“ auf Seite 158.

- Objekte, die die `c8y_Kpi`-Eigenschaft enthalten, werden jetzt (zusätzlich zu `pas_broadcastDevice`) als Broadcast-Geräte behandelt. Weitere Informationen finden Sie unter „Broadcast-Geräte“ auf Seite 136.
- Die Benutzeroberfläche von Analytics Builder ist nun in den Standardsprachen verfügbar, die von Cumulocity IoT unterstützt werden. Weitere Informationen finden Sie unter Verfügbare Sprachen im deutschsprachigen Cumulocity IoT *User guide*.
- Eine detaillierte Dokumentation steht nun zur Verfügung, in der die Unterschiede zwischen den Werttypen beschrieben werden, die kontinuierliche Zeitwerte und diskrete Zeitwerte darstellen. Außerdem finden Sie Informationen zum `pulse`-Typ, zu Zeitfenstern bei der Implementierung von Blöcken und wann Blöcke Ausgaben generieren. Siehe „Details zu Werten und Blöcken“ auf Seite 117.

Was ist neu in Version 10.7.0

- Es ist jetzt möglich, Vorlagenmodelle zu verwenden. Ein Vorlagenmodell ist ein Modell, in dem ein oder mehrere Vorlagenparameter definiert sind. Vorlagenparameter können an eine beliebige Anzahl von Blockparametern gebunden werden, vorausgesetzt, dass der Typ des Blockparameters mit dem des Vorlagenparameters übereinstimmt.

  Die folgenden Erweiterungen stehen mit dieser neuen Funktion zur Verfügung:

  - In einem Block können Sie einen Parameter zu einem Vorlagenparameter machen und anschließend einen neuen oder vorhandenen Vorlagenparameter zuweisen. Weitere Informationen finden Sie unter „Die Parameter eines Blocks editieren“ auf Seite 69.
  - Für die Bearbeitung der Vorlagenparameter steht ein neues Dialogfeld zur Verfügung. Weitere Informationen finden Sie unter „Vorlagenparameter verwalten“ auf Seite 78.
  - Ein neuer Editor, der Instanzeditor, steht jetzt zur Verfügung. Mit diesem Editor können Sie verschiedene Instanzen desselben Modells erstellen. Die Blöcke in jeder Instanz können dann unterschiedliche Werte für die Vorlagenparameter verwenden. Jede Instanz kann unabhängig von den anderen Instanzen aktiviert und deaktiviert werden, und jede Instanz kann in einem anderen Modus ausgeführt werden. Weitere Informationen finden Sie unter „Mit dem Instanzeditor arbeiten“ auf Seite 91.
  - Im Modellmanager gibt es jetzt eine neue Art von Karten für die Vorlagenmodelle. Weitere Informationen finden Sie unter „Die Benutzeroberfläche des Modellmanagers“ auf Seite 50.

  Siehe auch „Modelle“ auf Seite 40. In diesem Abschnitt wird der Unterschied zwischen den beiden Modellarten erläutert, die jetzt erstellt werden können: Modelle mit Vorlagenparametern und (wie in den vorherigen Versionen) Modelle ohne Vorlagenparameter. In diesem Abschnitt werden auch die relevanten Rollen für die neue Modellart erläutert: Modellautor und Instanzenverwalter. Diese Rollen können von derselben Person oder von verschiedenen Personen ausgeübt werden.

- Es können jetzt mehrere Geräte in einem Modell verwendet werden, das einen Gleichzeitigkeitsgrad von mehr als 1 hat. Siehe auch „Modellausführung für verschiedene Geräte“ auf Seite 134.
- Der Umgang mit alten Ereignissen ist jetzt nachsichtiger. Alte Ereignisse werden nur dann verworfen, wenn ein Ereignis für denselben Zeitstempel oder einen neueren Zeitstempel bereits von dem Modell verarbeitet wurde. Dies bedeutet, dass ein altes Ereignis möglicherweise von einigen Modellen verarbeitet, aber von anderen Modellen verworfen wird. Siehe auch „Eingabeblöcke und Ereignissteuerung“ auf Seite 112.
- Für langsame Ketten (englisch: „slow chains“) werden jetzt die folgenden Statusinformationen im Parameter `apama_status` veröffentlicht:
  - `user-analyticsbuilder.slowestChain.models`
  - `user-analyticsbuilder.slowestChain.delaySec`

  Zusätzlich wird jetzt eine Nachricht wird in das Korrelatorlog geschrieben, wenn die langsamste Kette um mehr als 1 Sekunde verzögert ist. Weitere Informationen finden Sie unter „Regelmäßig wiederkehrenden Status überwachen“ auf Seite 150.
- Aktivierungsmeldungen werden jetzt immer dann in das Korrelatorlog geschrieben, wenn die Aktivierung einer Kette gestartet und abgeschlossen wird. Weitere Informationen finden Sie unter „Den Lebenszyklus des Modell überwachen“ auf Seite 155.
- Die API-Version 1 des Analytics Builder Block SDK ist für Eingabe- und Ausgabeblöcke veraltet. Informationen zum Migrieren Ihrer benutzerdefinierten Eingabe- und Ausgabeblöcke auf die API-Version 2 finden Sie (auf Englisch) in der Dokumentation unter https://github.com/SoftwareAG/apama-analytics-builder-block-sdk.

  Wichtig:

  Die neuen Funktionen für mehrere Geräte in einem Modell und die nachsichtigere Behandlung alter Ereignisse werden deaktiviert, wenn Eingabe- und Ausgabeblöcke als Extensionen hinzugefügt werden, die die API-Version 1 des Analytics Builder Block SDK verwenden.
- Die Dokumentation enthält jetzt einen neuen Abschnitt, der neuen Benutzern einen kurzen Überblick darüber gibt, wie man ein einfaches Modell erstellen und sich dessen Ausgabe anzeigen lassen kann. Siehe „Erste Schritte: Erstellen Sie Ihr erstes Modell“ auf Seite 29.
- Die im Modelleditor zur Verfügung stehende Blockreferenz ist jetzt auch in der Dokumentation enthalten. Die „Übersicht aller Blöcke“ auf Seite 164 enthält Links zu den Beschreibungen aller Blöcke in der Blockreferenz.
- Analytics Builder unterstützt jetzt das Branding. Dies bedeutet, dass die im *User guide* von Cumulocity IoT beschriebenen Branding-Einstellungen jetzt auch für Analytics Builder gelten (siehe https://cumulocity.com/guides/users-guide/enterprise-edition/#branding; hiervon gibt es auch eine deutsche Version).
- Das Erscheinungsbild mehrerer Schaltflächen in der Symbolleiste des Modelleditors hat sich geändert. Insbesondere die folgenden Schaltflächen sehen jetzt anders aus: die Schaltfläche zum Ersetzen von Geräten sowie die Schaltfläche zum Anzeigen oder Ausblenden des Rasters.

Was ist neu in Version 10.6.6

- Der Eingabeblocktyp **Managed-Object-Eingabe** ist jetzt in der Lage den ursprünglichen Wert der Eigenschaft auszugeben, der beim Aktivieren des Modells mit dem Parameter **Eigenschaftsname** angegeben wurde. Zu diesem Zweck ist das Kontrollkästchen **Startwert erfassen** jetzt im Blockparametereditor verfügbar. Es ist standardmäßig deaktiviert (`false`). Ausführliche Informationen finden Sie in der Blockreferenz im Modelleditor.
- Mit dem **Ausdruck**-Block können jetzt Integer als Zwischenergebnisse ausgegeben werden, zum Beispiel als Ergebnis einer `floor()`- oder `round()`-Funktion. Wenn das Ergebnis eines Ausdrucks ein `integer`-Wert ist, dann wird er automatisch nach `float` konvertiert (dies könnte zu einem Präzisionsverlust führen). In früheren Versionen konnte der **Ausdruck**-Block nur die Typen `float`, `string` und `boolean` verarbeiten. Ausführliche Informationen finden Sie in der Blockreferenz im Modelleditor.

Was ist neu in Version 10.6

- Die folgenden neuen Blöcke stehen jetzt im Modelleditor zur Verfügung:
  - **Machine Learning** - Ruft das angegebene Machine Learning-Modell auf, das die Eingabedaten bewertet. Um diesen Block zu verwenden, muss die Machine Learning-Anwendung mit den entsprechenden Machine Learning-Modellen im Mandanten zur Verfügung stehen.
  - **Zähler** - Zählt die Gesamteingaben und die wiederholten Eingaben.

  Ausführliche Informationen zu den neuen Blöcken finden Sie in der Blockreferenz im Modelleditor.
- Der Blocktyp **Ereignisausgabe** hat jetzt einen zusätzlichen Eingangsport mit dem Namen **Texteingabe**. Wenn dieser Port verbunden ist, dann bestimmt er den Text des Ereignisses. Wenn der neue Eingangsport nicht verbunden ist, dann wird der Text mit dem Parameter **Nachricht** bestimmt. Wenn keines von beiden gesetzt ist, dann wird der Modellname als Text benutzt. Ausführliche Informationen zu dem neuen Eingangsport finden Sie in der Blockreferenz im Modelleditor.
- Analytics Builder unterstützt jetzt auch Polnisch. Siehe auch „Spracheinstellungen“ auf Seite 29.

Was ist neu in Version 10.5.7

- Die Dokumentation erwähnt jetzt die Berechtigungen, die für die Benutzung von Analytics Builder in Cumulocity IoT erforderlich sind. Siehe „Voraussetzungen“ auf Seite 28.
- Die folgenden Mandantenoptionen können jetzt auch für Messungen verwendet werden: `status_send_type` und `status_event_type`. Außerdem steht nun die neue Mandantenoption `status_send_keys` zur Verfügung. Siehe „Keys für die Statusberichterstattung“ auf Seite 157.
- Die Dokumentation wurde mit den für den Simulationsmodus verfügbaren Mandantenoptionen aktualisiert. Siehe „Keys für den Simulationsmodus“ auf Seite 159.

- Für Benutzer mit der LESEN-Berechtigung für "Regeln" (englisch: CEP management) stehen jetzt Diagnoseinformationen zur Verfügung. Diese Informationen umfassen den Inhalt der Logdatei, Kopien der EPL-Anwendungen und vieles mehr. Siehe „Diagnoseinformationen anzeigen“ auf Seite 156.
- Modellaktivierungen und -deaktivierungen werden in den Audit-Logs von Cumulocity IoT angezeigt. Siehe auch „Überwachungsprotokolle (Audit-Logs) anzeigen“ auf Seite 156.

Was ist neu in Version 10.5

- Sie können Analytics Builder jetzt mit Cumulocity IoT Core verwenden. Dies ist die Version von Cumulocity IoT, die in der Cloud verfügbar ist. In früheren Versionen war es nur möglich, Analytics Builder mit Cumulocity IoT Edge, der lokalen Version von Cumulocity IoT, zu verwenden.
- Im Modelleditor wird nun die hierarchische Struktur der Geräte und Gerätegruppen wiedergegeben. Dies ist in der Palette in den Kategorien **Eingabe** und **Ausgabe** zu sehen, im Blockparametereditor bei der Auswahl eines anderen Geräts, und beim Ersetzen von Geräten.
  - Wenn Sie das Suchfeld in der Palette des Modelleditors verwenden (verfügbar für Eingabe- und Ausgabeblöcke), werden jetzt alle Assets im Cumulocity IoT-Inventar angezeigt, die Ihren Suchkriterien entsprechen (siehe „Einen Block hinzufügen“ auf Seite 67).
  - Wenn Sie im Blockparametereditor ein anderes Gerät auswählen oder wenn Sie Geräte ersetzen, wird jetzt ein Dialogfeld angezeigt, in dem Sie das gewünschte Gerät oder die gewünschte Gerätegruppe auswählen können. Wenn Sie das Suchfeld in diesem Dialogfeld verwenden, können Sie auch ein beliebiges anderes Asset aus dem Cumulocity IoT-Inventar auswählen, das Ihren Suchkriterien entspricht. Siehe auch „Die Parameter eines Blocks editieren“ auf Seite 69 und „Geräte, Gerätegruppen und Assets ersetzen“ auf Seite 76
  - Mit einer Mandantenoption können Sie die Suche so einschränken, dass nur Assets eines bestimmten Typs (zum Beispiel, nur Geräte) angezeigt werden (siehe „Nach Eingabe- und Ausgabe-Assets suchen“ auf Seite 141).
- Die folgenden neuen Blocktypen stehen jetzt im Modelleditor zur Verfügung:
  - **Alarmeingabe** - Eingabeblock für die Verarbeitung empfangener `Alarm`-Objekte.
  - **Alarmausgabe** - Ausgabeblock zum Erstellen eines neuen `Alarm`-Objekts.
  - **Ereignisausgabe** - Ausgabeblock zum Erstellen eines neuen `Event`-Objekts.
  - **Managed-Object-Eingabe** - Eingabeblock für die Verarbeitung empfangener `ManagedObject`-Objekte.
  - **Managed-Object-Ausgabe** - Ausgabeblock zum Aktualisieren einer vorkonfigurierten Eigenschaft des `ManagedObject`-Objekts.
  - **Operationseingabe** - Eingabeblock für die Verarbeitung empfangener `Operation`-Objekte.

  Ausführliche Informationen zu den neuen Blocktypen finden Sie in der Blockreferenz im Modelleditor.

- Die Eingabeblocktypen **Messwerteingabe** und **Ereigniseingabe** sind jetzt in der Lage den Zeitstempel des eingehenden Messwertes oder Ereignisses zu ignorieren. Zu diesem Zweck ist das Kontrollkästchen **Zeitstempel ignorieren** jetzt im Blockparametereditor verfügbar. Es ist standardmäßig deaktiviert (`false`). Ausführliche Informationen finden Sie in der Blockreferenz im Modelleditor.
- Die Ausgabeblöcke unterstützen jetzt zeitasynchrone Ausgaben. Der Ausgabeblocktyp **Operationsausgabe** ist jetzt in der Lage asynchrone Ausgaben zu erzeugen. Siehe auch „Ausgabeblöcke und Ereignissteuerung“ auf Seite 113.
- Es ist jetzt möglich, mehrere Ausgabeblöcke des Typs **Operationsausgabe** in einem einzelnen Modell oder in mehreren Modellen zu verwenden, um neue Operationen für dasselbe Gerät zu erstellen.
- Der **Ausdruck**-Block kann jetzt integrierte statische Methoden (englisch: „built-in static methods“) aufrufen und auf integrierte Konstanten (englisch: „built-in constants“) für die unterstützten Eingabetypen zugreifen. Ausführliche Informationen finden Sie in der Blockreferenz im Modelleditor.
- Der **Schwelle**-Block unterstützt jetzt weitere Optionen für den Parameter **Richtung**: **Oberhalb oder gleich** und **Unterhalb oder gleich**. Ausführliche Informationen finden Sie in der Blockreferenz im Modelleditor.
- Im Modelleditor ist es jetzt möglich, einen Block aus der Palette mit Drag & Drop in eine aufgeklappte Gruppe auf der Arbeitsfläche zu ziehen. In der vorherigen Version wurden die Gruppen auf der Arbeitsfläche beim Ziehen eines Blocks aus der Palette ignoriert, und der Block wurde stattdessen auf der Arbeitsfläche (unter der Gruppe) abgelegt. Siehe auch „Einen Block hinzufügen“ auf Seite 67.
- Cumulocity IoT Edge wird jetzt genauso konfiguriert wie Cumulocity IoT Core. Es ist daher nicht mehr möglich, dies mit Konfigurationsdateien in `/usr/edge/properties/apama/extensions/config/files/` zu tun; diese Möglichkeit wurde entfernt. Siehe auch „Konfiguration“ auf Seite 157.
- Sie können jetzt verschiedene Mandantenoptionen festlegen oder ändern, indem Sie REST-Anfragen an Cumulocity IoT senden. Ausführliche Informationen finden Sie unter „Konfiguration“ auf Seite 157.
- Die Statusberichterstattung ist jetzt standardmäßig deaktiviert. Siehe „Konfiguration“ auf Seite 157; dort wird erklärt, wie Sie dies mit `status_period_secs` und `status_device_name` ändern können.
- Der Standardname des Cumulocity IoT-Geräts, auf dem die Statusoperationen veröffentlicht werden sollen (`status_device_name`) wurde von `c8y_EdgeGateway` auf `apama_status` abgeändert. Siehe auch „Keys für die Statusberichterstattung“ auf Seite 157..
- Der Standardwert für den Gleichzeitigkeitsgrad (`numWorkerThreads`) wurde von 4 auf 1 abgeändert. Siehe auch „Den Gleichzeitigkeitsgrad definieren“ auf Seite 136.
- Es ist nicht mehr möglich, den Korrelator mit den Befehlszeilen-Tools (den sogenannten „command-line tools“) von Apama zu überwachen.

- Die Unterstützung für das Erstellen von Erweiterungen wurde aus Analytics Builder entfernt. Statt dessen können Sie Erweiterungen jetzt mit dem Analytics Builder Block SDK erstellen.
- Mit dem Analytics Builder Block SDK können Sie jetzt Ihre eigenen, benutzerdefinierten Blöcke schreiben, testen, verpacken und nach Analytics Builder hochladen. Das Block SDK finden Sie in GitHub unter https://github.com/SoftwareAG/apama-analytics-builder-block-sdk. Siehe auch „Eigene Blöcke erstellen“ auf Seite 45.

Was ist neu in Version 10.3.2

- Analytics Builder unterstützt jetzt Gerätegruppen, die im Cumulocity IoT-Inventar definiert wurden. Sie können in der Palette aus der Kategorie **Eingabe** ausgewählt werden. Siehe auch „Einen Block hinzufügen“ auf Seite 67.
- Wenn Sie die Kategorie **Eingabe** oder **Ausgabe** in der Palette aufklappen, wird jetzt ein Suchfeld angezeigt, mit dem Sie die benötigten Geräte und Gerätegruppen anzeigen können. Bei der Suche wird zwischen Groß- und Kleinschreibung unterschieden. Weitere Informationen finden Sie unter „Einen Block hinzufügen“ auf Seite 67.

  Anmerkung:

  Analytics Builder unterstützt nur Geräte, die im Cumulocity IoT-Inventar mit einem `c8y_IsDevice`-Fragment markiert wurden. Kindgeräte werden nicht unterstützt.
- Die Anzahl der Geräte (und Gerätegruppen), die in der Palette angezeigt werden, kann jetzt angepasst werden. Ausführliche Informationen finden Sie unter „Die Anzahl der angezeigten Geräte, Gerätegruppen und/oder Assets konfigurieren“ auf Seite 141.
- Ein spezieller Ausgabeblock mit dem Namen **Trigger-Gerät** kann jetzt aus der Palette ausgewählt werden. Dieser Block schickt die Ausgabe zurück an das Gerät, das die Ausgabe ausgelöst hat. Modelle können jetzt Daten von mehreren Geräten verarbeiten und dabei (bei Verwendung von mehreren Kernen) skalieren. Ausführliche Informationen finden Sie unter „Modellausführung für verschiedene Geräte“ auf Seite 134.
- Analytics Builder unterstützt jetzt Eingabegeräte, die als „Broadcast-Geräte“ bezeichnet werden. Die Signale von diesen Geräten stehen allen Modellen und Geräten zur Verfügung. Ausführliche Informationen finden Sie unter „Broadcast-Geräte“ auf Seite 136.
- In früheren Versionen von Analytics Builder war es möglich, ein Modell zu erstellen, das Daten von mehreren Geräten empfängt oder Daten an mehrere Geräte sendet. Dies ist mit der Standardkonfiguration nicht mehr möglich. Der Versuch, ein solches Modell zu aktivieren, resultiert in einem Fehler. Damit Sie solche Modelle ausführen können, müssen Sie den Gleichzeitigkeitsgrad auf 1 setzen (siehe „Modellausführung für verschiedene Geräte“ auf Seite 134). Dies schränkt die Leistung bei der Ausführung von Analysen ein.
- Die Beschriftungen der Blöcke auf der Arbeitsfläche des Modelleditors enthalten jetzt den Wert des wichtigsten Parameters:
  - Der **Ausdruck**-Block zeigt jetzt den definierten Ausdruck.
  - Der **Schwelle**-Block zeigt jetzt den definierten Schwellenwert.

  - Der **Kombinierer**-Block zeigt jetzt den definierten Modus.
  - Der **Eigenschaft extrahieren**-Block zeigt jetzt den definierten Eigenschaftsnamen.
  - Der **Rundung**-Block zeigt jetzt die definierte Regel.
  - Der **Zeitverzögerung**-Block zeigt jetzt die definierte Verzögerung in Sekunden.
- Der **Ausdruck**-Block behandelt boolesche Literale und logische Operatoren jetzt unabhängig von Groß- und Kleinschreibung. Bei der "Event Processing Language" (EPL) von Apama ist dies anders; EPL akzeptiert bei booleschen Literalen und logischen Operatoren nur Kleinschreibung.
- Die Schaltfläche zum Verlassen des Modelleditors befindet sich jetzt ganz rechts in der Symbolleiste. Siehe auch „Den Modelleditor verlassen“ auf Seite 67.
- Sie können jetzt Aspekte des Korrelators mit den Befehlszeilen-Tools (den sogenannten „command-line tools“) von Apama überwachen, einschließlich der Warteschlangengrößen und Zugriff auf die Logdatei.
- Sie können Analytics Builder jetzt mit Hilfe von Erweiterungen anpassen.
- Die folgenden Pfade werden jetzt benutzt (jetzt mit `extensions/config/files` im Pfad):
  - `/usr/edge/properties/apama/extensions/config/files/support/cumulocity/`
  - `/usr/edge/properties/apama/extensions/config/files/framework/monitors/`

  Anmerkung:

  Der Speicherort der Konfigurationsdateien kann sich in zukünftigen Hauptversionen (englisch: major releases) ändern.
- Die Dokumentation enthält jetzt Informationen zu virtuellen Geräten. Siehe „Virtuelle Geräte“ auf Seite 137.

Was ist neu in Version 10.3.1

- Der **Ausdruck**-Block unterstützt jetzt auch die `string`- und `boolean`-Typen für Eingabe- und Ergebniswerte. Logische Operatoren können jetzt für boolesche Werte verwendet werden. Die Unterstützung für die vorhandenen relationalen, numerischen und Gleichheitsoperatoren wurde da wo es angebracht ist auf `string`- und `boolean`-Typen erweitert. Ausführliche Informationen zur neuen Funktionalität finden Sie in der Blockreferenz im Modelleditor.
- Der **Überschreitungszähler**-Block kann jetzt mit einem Zeitfenster arbeiten, das mit dem Parameter **Fensterdauer** angegeben wird. Ausführliche Informationen zur neuen Funktionalität finden Sie in der Blockreferenz im Modelleditor.
- Ein neuer **Standardabweichung**-Block steht jetzt zur Verfügung. Mit diesem Block können Sie die Standardabweichung und die Varianz der Werte im Laufe der Zeit berechnen. Ausführliche Informationen zu diesem Block finden Sie in der Blockreferenz im Modelleditor.

- Ein neuer **Kombinierer**-Block steht jetzt zur Verfügung. Mit diesem Block können Sie die Ausgabe auf Grundlage des gewählten Modus und der verbundenen Eingänge berechnen. Ausführliche Informationen zu diesem Block finden Sie in der Blockreferenz im Modelleditor.
- Ein neuer **Gradient**-Block steht jetzt zur Verfügung. Mit diesem Block können Sie den gewichteten linearen Regressionsgradienten für die Werte berechnen. Ausführliche Informationen zu diesem Block finden Sie in der Blockreferenz im Modelleditor.
- Die Blöcke, Verbindungen und Gruppen auf der Arbeitsfläche werden jetzt immer an einem Raster ausgerichtet. Sie können entscheiden, ob das Raster angezeigt werden soll oder nicht. Standardmäßig wird es nicht angezeigt. Siehe auch „Das Raster anzeigen und ausblenden“ auf Seite 89.
- Die Benutzeroberfläche von Analytics Builder kann jetzt auch in Deutsch angezeigt werden. Siehe auch „Spracheinstellungen“ auf Seite 29.

2 Einführung in Analytics Builder

- Was ist Analytics Builder? 26
- Analytics Builder und Cumulocity IoT 26
- Voraussetzungen 28
- Analytics Builder starten 28
- Spracheinstellungen 29
- Erste Schritte: Erstellen Sie Ihr erstes Modell 29
- Erste Schritte: Erstellen Sie ein Modell aus einem Beispiel 33

Was ist Analytics Builder?

Analytics Builder ist Teil der Streaming Analytics-Anwendung, die in Cumulocity IoT ausgeführt wird, einer webbasierten Plattform zur Verwaltung von IoT-Geräten (siehe auch https://www.cumulocity.com/). Mit Analytics Builder können Sie Analysemodelle erstellen, die Streaming-Daten transformieren oder analysieren und somit neue Daten oder Ausgabeereignisse generieren. Die Modelle sind in der Lage, die Daten in Echtzeit zu verarbeiten.

Sie entwerfen die Modelle in einer grafischen Umgebung, indem Sie vorgefertigte *Blöcke* zu *Modellen* kombinieren. Die Blöcke in einem Modell bündeln kleinere logische Elemente und haben eine Reihe von Eingängen, Ausgängen und Parametern. Jeder Block implementiert eine bestimmte Funktionalität, wie zum Beispiel das Empfangen von Daten von einem Sensor, das Durchführen einer Berechnung, das Erkennen eines Zustandes oder das Erzeugen eines Ausgangssignals. Die Konfiguration der Blöcke wird von Ihnen definiert und Sie verknüpfen die Blöcke mit Hilfe von sogenannten *Verbindungen*. Sie können die Modelle editieren, die Ausführung eines Modells mit historischen Daten simulieren oder die Modelle in der Produktionsumgebung ausführen. Weitere Informationen finden Sie unter „Modelle verstehen“ auf Seite 39.

Analytics Builder besteht aus den folgenden Tools:

- **Modellmanager.** Wenn Sie Analytics Builder aufrufen, wird zuerst der Modellmanager angezeigt. Alle verfügbaren Modelle werden hier aufgelistet und Sie können die Modelle verwalten. Mit dem Modellmanager können Sie die Modelle zum Beispiel testen und für die Produktion aktivieren, oder Sie können die Modelle duplizieren oder entfernen. Sie können auch neue Modelle erstellen oder vorhandene Modelle bearbeiten; in diesen Fällen wird der Modelleditor aufgerufen. Es stehen auch Beispiele zur Verfügung, die Ihnen den Einstieg in die Erstellung eigener Modelle erleichtern sollen. Weitere Informationen finden Sie unter „Mit dem Modellmanager arbeiten“ auf Seite 49.
- **Modelleditor.** Mit dem Modelleditor definieren Sie alle Blöcke, die in einem Modell benutzt werden sollen, und Sie verbinden die Blöcke miteinander. Die Funktionalität der verschiedenen Blöcke ist im Modelleditor in der sogenannten *Blockreferenz* beschrieben. Weitere Informationen finden Sie unter „Mit dem Modelleditor arbeiten“ auf Seite 63.
- **Instanzeditor.** Wenn in einem Modell Vorlagenparameter definiert wurden, können Sie mit dem Instanzeditor verschiedene Instanzen desselben Modells erstellen, die dann separat aktiviert und verwaltet werden können. Der Instanzeditor verwendet die Vorlagenparameter, die im Modelleditor definiert wurden. Weitere Informationen finden Sie unter „Mit dem Instanzeditor arbeiten“ auf Seite 91.

Die Blöcke sind in der "Event Processing Language" (EPL) von Apama implementiert. Zum Ausführen der Modelle wird der EPL-Code zur Laufzeit in einem Apama-Korrelator ausgeführt. Es ist wichtig, dass Sie das Laufzeitverhalten und die Einschränkungen verstehen. Dies ist in späteren Kapiteln beschrieben.

Analytics Builder und Cumulocity IoT

Cumulocity IoT ist eine Plattform zum Verbinden, Überwachen und Steuern von Remote-Geräten. Einen Überblick finden Sie im *Concepts guide* unter https://www.cumulocity.com/guides/.

Geräte und Sensoren können mit Cumulocity IoT verbunden werden. Siehe die Informationen über Geräteschnittstellen im *Concepts guide* und die Informationen zur Geräteintegration mit MQTT im *Device SDK guide*, beides unter der oben genannten URL.

Sensoren schicken `Measurement`- oder `Event`-Objekte an Cumulocity IoT, und die Geräte können `Operation`-Objekte empfangen, die innerhalb der Cumulocity IoT-Plattform erstellt wurden. Alle diese Objekte (`Measurement`, `Event`, `Operation`) werden einem einzelnen Gerät auf der Cumulocity IoT-Plattform zugeordnet. Einem Gerät können mehrere Messtypen zugeordnet sein; und die Messtypen, die von einem Gerät unterstützt werden, können dieselben sein wie bei anderen Geräten oder anders als bei anderen Geräten. Sobald ein Gerät mit Cumulocity IoT verbunden ist, werden die Informationen über dieses Gerät im Cumulocity IoT-Inventar gespeichert. Diese sind in der Device Management-Anwendung sichtbar, welche auch dazu verwendet werden kann, die `Measurement`-, `Event`- oder `Operation`-Objekte anzuzeigen, die mit diesem Gerät verknüpft sind. Weitere Informationen zur Geräteverwaltung finden Sie im *User guide* (hiervon gibt es auch eine deutsche Version) unter der oben genannten URL.

Die Cumulocity IoT-Plattform enthält eine Apama-Korrelatorkomponente, die von der Cumulocity IoT-Plattform verwaltet wird (diese wird nicht manuell gestartet oder gestoppt) und die für die Kommunikation mit Cumulocity IoT vorkonfiguriert ist. Dieser Korrelator beherbergt die Analytics Builder-Laufzeit und führt auch alle benutzerdefinierten Apama-Regeln aus, die mit EPL-Apps hinzugefügt wurden. Weitere Informationen finden Sie im *Streaming Analytics guide* unter der oben genannten URL.

Mit Analytics Builder können Sie Modelle erstellen, die mit Geräten und Sensormessungen interagieren. Modelle können `Measurement`- und `Event`-Objekte von Geräten empfangen; diese enthalten die Eingaben für Berechnungen oder Mustererkennungen, die mit dem Modell durchgeführt werden. Modelle können neue `Measurement`-Objekte erstellen, die von Sensoren hergeleitete Werte darstellen können (zum Beispiel, eine Durchschnittstemperatur), oder die Messwerte können als Eingabe für andere Analysemodelle verwendet werden (siehe „Verbindungen zwischen Modellen“ auf Seite 138). Modelle können neue `Operation`-Objekte erstellen, die zur Gerätesteuerung an die Geräte gesendet werden (zum Beispiel, um einen Alarm ertönen zu lassen, eine Nachricht auf einem Bildschirm anzuzeigen oder ein Gerät auszuschalten). Die Modelle werden auch im Cumulocity IoT-Inventar gespeichert, können jedoch mit dem Modellmanager importiert oder exportiert werden.

Geschäftslogik kann auch in Apamas Event Processing Language (Apama EPL) geschrieben werden. Apama EPL bietet mehr Leistung und Flexibilität in einer textbasierten Programmiersprache. Dies ist eine Alternative, wenn komplexere Logik erforderlich ist oder die Logik nicht in das Muster eines Analysemodells passt. EPL-Apps können direkt in Cumulocity IoT mit der Streaming Analytics-Anwendung geschrieben werden. Weitere Informationen, einschließlich Beispielen, finden Sie im *Streaming Analytics guide* unter https://www.cumulocity.com/guides/. Alternativ ist es auch möglich, benutzerdefinierte Blöcke zu erstellen, wenn keiner der vorgefertigten Blöcke von Analytics Builder die erforderliche Logik implementiert; siehe „Eigene Blöcke erstellen“ auf Seite 45.

Analytics Builder kann sowohl mit Cumulocity IoT Core (Cloud) als auch mit Cumulocity IoT Edge (lokale Installation) verwendet werden. Sie können verschiedene Aspekte von Analytics Builder anpassen, indem Sie verschiedene Mandantenoptionen setzen. Ausführliche Informationen finden Sie unter „Konfiguration“ auf Seite 157.

Voraussetzungen

Browser

Analytics Builder unterstützt dieselben Browser wie Cumulocity IoT, mit der folgenden Ausnahme: Browser auf Smartphones und Tablets werden nicht unterstützt.

Berechtigungen

Damit Sie Analytics Builder in Cumulocity IoT verwenden können, benötigen Sie mindestens die folgenden Berechtigungen:

Berechtigungstyp	Berechtigungsebene
Regeln (englisch: CEP management)	ADMIN
Optionen (englisch: Option management)	LESEN
Stammdaten (englisch: Inventory)	LESEN

Dies wird normalerweise durch die Verwendung einer globalen Rolle erreicht, die über diese Berechtigungen verfügt, und wobei die Rolle Zugriff auf die Anwendung Analytics Builder hat. Einzelheiten zum Verwalten von Berechtigungen finden Sie bei den Informationen zur Administrationsanwendung im *User guide* unter https://www.cumulocity.com/guides/ (hiervon gibt es auch eine deutsche Version).

Microservice

Für die uneingeschränkte Arbeit mit Analytics Builder benötigen Sie den Apama-ctrl-Microservice in Cumulocity IoT.

Falls der Apama-ctrl-starter-Microservice auf Ihrem Mandanten benutzt wird, dann ist Analytics Builder auf maximal 3 aktive Modelle beschränkt. Mit dem Analytics Builder Block SDK geschriebene, benutzerdefinierte Blöcke können mit Apama-ctrl-starter nicht verwendet werden. Wenden Sie sich an den Support der Software AG, wenn Sie weitere Funktionen benötigen.

Falls der Apama-ctrl-smartrules-Microservice auf Ihrem Mandanten benutzt wird, dann steht Analytics Builder nicht zur Verfügung.

Analytics Builder starten

Analytics Builder ist Teil der Streaming Analytics-Anwendung in Cumulocity IoT. Sie starten diese Anwendung mit dem Application Switcher, den Sie in der oberen Leiste der Benutzeroberfläche von Cumulocity IoT finden können.

Ausführliche Informationen zur Benutzung von Cumulocity IoT finden Sie unter https://www.cumulocity.com/guides/.

Fragen Sie Ihren Administrator nach der URL, die zum Starten von Cumulocity IoT erforderlich ist.

Spracheinstellungen

Die Sprache, in der die Benutzeroberfläche von Analytics Builder angezeigt wird, hängt von Ihren Benutzereinstellungen in Cumulocity IoT ab. Weitere Informationen finden Sie im *User guide* (hiervon gibt es auch eine deutsche Version) unter https://www.cumulocity.com/guides/.

Wenn Cumulocity IoT oder der Browser auf eine Sprache eingestellt ist, die zur Zeit nicht von Analytics Builder unterstützt wird, dann wird die Benutzeroberfläche mit der Standardsprache Englisch angezeigt.

Erste Schritte: Erstellen Sie Ihr erstes Modell

Dieser Abschnitt gibt einen kurzen Überblick darüber, wie Sie ein neues Modell erstellen und dessen Ausgabe anzeigen können. Er ist nicht als umfassende Beschreibung aller Möglichkeiten von Analytics Builder gedacht. Daher sind die Erklärungen auf ein Minimum beschränkt. Ausführlichere Informationen finden Sie im Rest dieser Dokumentation.

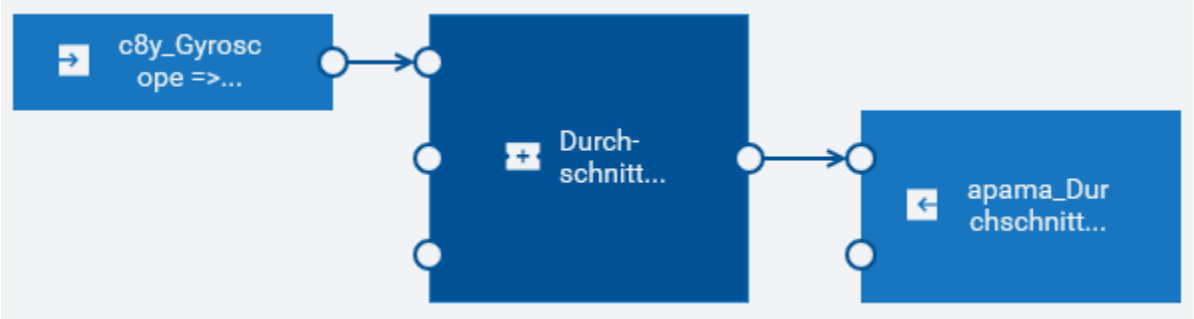
Die folgenden Schritte setzen voraus, dass bereits ein Gerät in Cumulocity IoT registriert wurde. Vorzugsweise ist dies ein Gerät, das bereits Messwerte an Cumulocity IoT sendet. Dies kann zum Beispiel ein Smartphone sein, auf dem die Cumulocity IoT Sensor App installiert ist (detaillierte Informationen finden Sie unter Cumulocity IoT Sensor App in der Cumulocity IoT-Dokumentation).

Das Modell, das Sie erstellen, wird drei Blöcke enthalten:

- Einen Eingabeblock, der Messwerte von einem Gerät empfängt.

- Einen Block, der den Mittelwert der Messwerte im Laufe der Zeit berechnet.
- Einen Ausgabeblock, der die berechneten Mittelwerte an die Device Management-Anwendung von Cumulocity IoT sendet, damit sie dort angezeigt werden können.

Wenn Sie alle unten aufgeführten Schritte ausgeführt haben, dann wird Ihr Modell ähnlich wie das Folgende aussehen:



Schritt 1: Melden Sie sich bei Cumulocity IoT an und wechseln Sie zu Analytics Builder

Sie greifen über einen Webbrowser auf Cumulocity IoT zu. Ausführliche Informationen finden Sie unter Erste Schritte in der Cumulocity IoT-Dokumentation.

1. Fragen Sie Ihren Administrator nach der URL, die zum Starten von Cumulocity IoT erforderlich ist.
2. Geben Sie auf der Anmeldeseite Ihren Benutzernamen und Ihr Passwort ein.
3. Klicken Sie in der oberen Leiste der Cumulocity IoT-Benutzeroberfläche auf die Schaltfläche für den Application Switcher ( ) und dann auf **Streaming Analytics**.
4. Klicken Sie auf der Startseite der Streaming Analytics-Anwendung unter der Überschrift „Analytics Builder“ auf die Schaltfläche **Öffnen**.

   Oder klicken Sie auf der linken Seite im Navigator auf **Analytics Builder**.

   Anmerkung:

   Wenn der Navigator gerade verborgen ist, klicken Sie auf den kleinen Pfeil ganz links in der oberen Leiste um die Anzeige des Navigator einzublenden.

Schritt 2: Erstellen Sie ein neues Modell

Die erste Seite, die nach dem Aufruf von Analytics Builder angezeigt wird, ist der Modellmanager.

1. Auf der Registerkarte **Modelle** des Modellmanagers: klicken Sie in der Symbolleiste auf **Neues Modell**.
2. Geben Sie im anschließend angezeigten Dialogfeld einen Modellnamen ein und klicken Sie auf **OK**.

Schritt 3: Fügen Sie den Eingabeblock hinzu

Sie entwerfen Ihr Modell im Modelleditor. Nachdem Sie den Modellnamen eingegeben haben, wird der Modelleditor angezeigt. Die Palette auf der linken Seite enthält alle Blöcke, die Sie Ihrem Modell hinzufügen können. Sie fügen einen Block hinzu, indem Sie ihn von der Palette auf die Arbeitsfläche ziehen. Die Blöcke für die Eingabegeräte, die in Cumulocity IoT registriert wurden, werden in der Palette unter **Eingabe** angezeigt.

1. Ziehen Sie den Eingabeblock für Ihr Gerät auf die Arbeitsfläche.

   Der Blockparametereditor wird automatisch angezeigt. Der Blocktyp eines Eingabegeräts ist standardmäßig **Messwert-Eingabe**. Wir werden diese Standardeinstellung verwenden.

   Anmerkung:

   Wenn der Blockparametereditor nicht angezeigt wird (zum Beispiel, weil Sie auf eine leere Stelle der Arbeitsfläche geklickt haben, nachdem Sie den Eingabeblock auf die Arbeitsfläche gezogen haben), klicken Sie mit der *linken* Maustaste auf den Block, um den Blockparametereditor anzuzeigen.

2. Wählen Sie das Fragment und die Serie aus, worauf der Eingabeblock hören soll.

   Wenn Ihr Gerät zuvor Daten gesendet hat, bietet das Dropdown-Listenfeld einen oder mehrere Werte zur Auswahl an. Ein Beispiel für die Cumulocity IoT Sensor App wäre **c8y_Gyroscope => gyroscopeY**.

3. Markieren Sie das Kontrollkästchen **Zeitstempel ignorieren**.

   Dadurch wird sichergestellt, dass die Messwerte in der gleichen Reihenfolge verarbeitet werden, in der sie empfangen werden.

Lesen Sie die Blockreferenz in der Dokumentationsanzeige auf der rechten Seite, wenn Sie detaillierte Informationen zum aktuell ausgewählten Block benötigen. Wenn die Dokumentationsanzeige zur Zeit nicht sichtbar ist, klicken Sie .

Schritt 4: Fügen Sie den Block hinzu, der den Mittelwert der Messwerte berechnet

1. Klappen Sie in der Palette die Kategorie **Aggregat** auf.
2. Ziehen Sie den Block **Durchschnitt (Mittelwert)** auf die Arbeitsfläche.
3. Geben Sie im Blockparametereditor einen Wert für **Fensterdauer (Sek.)** an, zum Beispiel "10".

   Die angegebene Anzahl von Sekunden steuert die Dauer, über welche die Messung gemittelt wird. Kleinere Werte reagieren schneller auf Wertänderungen, größere Werte bewirken eine bessere Glättung des Wertes.

Schritt 5: Fügen Sie den Ausgabeblock hinzu

1. Klappen Sie in der Palette die Kategorie **Ausgabe** auf.

2. Ziehen Sie den Ausgabeblock für Ihr Gerät (das heißt, einen Block mit dem selben Namen wie Ihr Eingabeblock) auf die Arbeitsfläche.
3. Geben Sie "apama_Durchschnitt" als Fragmentname an.
4. Geben Sie "wert" als Name der Serie an.

Schritt 6: Verbinden Sie die Blöcke

Um die Werte von einem Block zum anderen zu übertragen, müssen Sie die Blöcke mit den sogenannten Verbindungen verknüpfen. De Verbindungen werden an den Ports angeschlossen, das heißt, an den kleinen Kreisen, die links und/oder rechts von einem Block angezeigt werden.

1. Klicken Sie auf den **Wert**-Ausgangsport des Eingabeblocks und ziehen Sie die Maus zum **Wert**-Eingangsport des **Durchschnitt (Mittelwert)**-Blocks.
2. Klicken Sie auf den **Durchschnitt**-Ausgangsport des **Durchschnitt (Mittelwert)**-Blocks und ziehen Sie die Maus zum **Wert**-Eingangsport des Ausgabeblocks.

Schritt 7: Speichern Sie das Modell und kehren Sie zum Modellmanager zurück

1. Klicken Sie in der Symbolleiste des Modelleditors auf um Ihr neu erstelltes Modell zu speichern.
2. Klicken Sie in der Symbolleiste des Modelleditors auf um den Modelleditor zu verlassen und zum Modellmanager zurückzukehren.

Anmerkung:

Auf der Registerkarte **Modelle** des Modellmanagers werden nur die Modelle angezeigt, die explizit gespeichert wurden. Wenn Sie ein neues Modell erstellen und den Modelleditor verlassen, ohne das Modell zu speichern, dann wird es nicht im Modellmanager angezeigt und all Ihre Änderungen sind verloren.

Schritt 8: Aktivieren Sie das Modell im Produktionsmodus

Auf der Registerkarte **Modelle** des Modellmanagers wird eine Karte für das neu erstellte Modell angezeigt. Ein neues Modell wird automatisch in den Entwurfsmodus und in den inaktiven Status versetzt. Sie werden nun Ihr neues Modell im Produktionsmodus aktivieren. Dadurch wird das Modell eingesetzt und die Messungen von Ihrem Gerät werden verarbeitet.

1. Klicken Sie auf der Karte auf das Dropdown-Menü, das derzeit **Entwurf** anzeigt und wählen Sie **Produktion**.
2. Klicken Sie auf der Karte auf die Umschaltfläche, die derzeit **Inaktiv** anzeigt. Dies ändert den Status in **Aktiv**.

Schritt 9: Gehen Sie zum Device Management und zeigen Sie die Messungen an

Um die von Ihrem aktiven Modell gesendeten Messungen anzuzeigen, müssen Sie zur Device Management-Anwendung von Cumulocity IoT wechseln. Ausführliche Informationen finden Sie unter Device Management in der Cumulocity IoT-Dokumentation.

1. Klicken Sie auf die Schaltfläche für den Application Switcher ( ) und dann auf **Device Management**.
2. Klicken Sie im Navigator auf der linken Seiten auf **Geräte** und dann auf **Alle Geräte**.
3. Lokalisieren Sie Ihr Gerät und klicken Sie auf dessen Name um die Gerätedetails anzuzeigen.
4. Klicken Sie links auf **Messwerte**. Dies ist eine dynamische Registerkarte, die nur dann angezeigt wird, wenn Messwerte für das Gerät vorhanden sind.

   Auf der daraufhin erscheinenden Seite werden mehrere Diagramme angezeigt, in denen die von Ihrem Gerät gesendeten Daten visualisiert werden. Es sollte jetzt auch ein Diagramm mit dem Titel "Apama_durchschnitt" angezeigt werden, in dem Sie die Werte sehen können, die von Ihrem neu erstellten Modell gesendet werden. Möglicherweise müssen Sie die Seite neu laden, damit dieses neue Diagramm angezeigt wird. Weitere Informationen über diese Registerkarte finden Sie unter Messwerte in der Cumulocity IoT-Dokumentation.

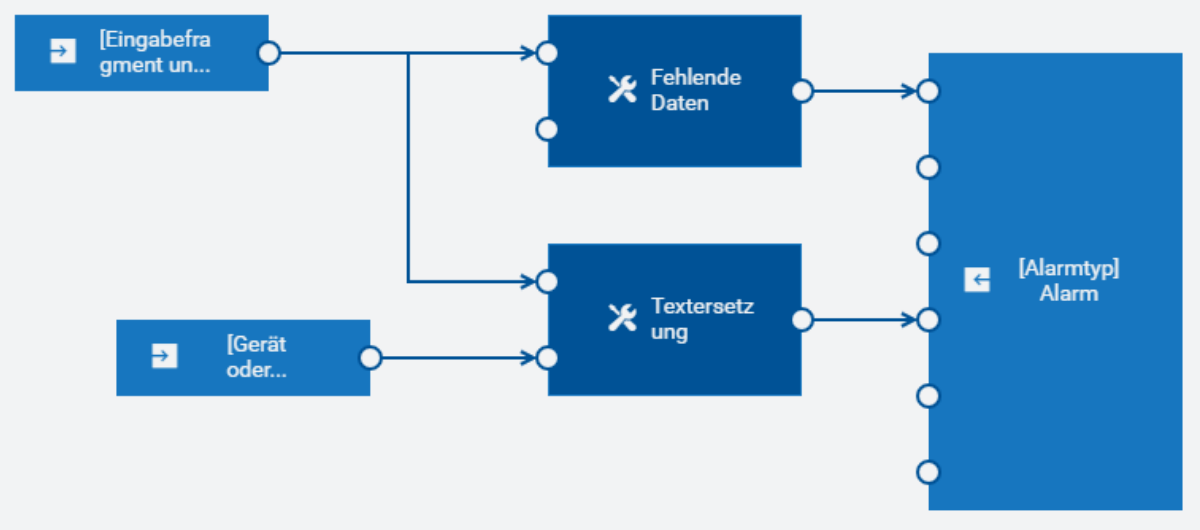
Erste Schritte: Erstellen Sie ein Modell aus einem Beispiel

Dieser Abschnitt gibt einen kurzen Überblick darüber, wie Sie ein Modell aus einem Beispiel erstellen können. Er basiert auf dem Beispiel mit dem Namen **Bei fehlenden Messdaten Alarm erzeugen**. Ihr neues Modell erzeugt einen Alarm, wenn innerhalb eines bestimmten Zeitraums keine neuen Messdaten eingehen.

Dieser Abschnitt ist nicht als umfassende Beschreibung aller Möglichkeiten von Analytics Builder gedacht. Daher sind die Erklärungen auf ein Minimum beschränkt. Ausführlichere Informationen finden Sie im Rest dieser Dokumentation.

Die folgenden Schritte setzen voraus, dass bereits ein Gerät in Cumulocity IoT registriert wurde. Vorzugsweise ist dies ein Gerät, das bereits Messwerte an Cumulocity IoT sendet. Dies kann zum Beispiel ein Smartphone sein, auf dem die Cumulocity IoT Sensor App installiert ist (detaillierte Informationen finden Sie unter Cumulocity IoT Sensor App in der Cumulocity IoT-Dokumentation).

Die folgende Abbildung zeigt die Blöcke, die im Beispiel **Bei fehlenden Messdaten Alarm erzeugen** definiert sind.



Das Beispiel verwendet vordefinierte Vorlagenparameter. Nachdem Sie ein Modell aus dem Beispiel erstellt haben, können Sie mehrere Instanzen des Modells erstellen und unterschiedliche Werte für die Vorlagenparameter angeben. Siehe auch „Modelle“ auf Seite 40 für eine Erklärung des Unterschieds zwischen Modellen ohne Vorlagenparametern und Modellen mit Vorlagenparametern.

Im Folgenden finden Sie eine kurze Beschreibung der Blöcke, die in dem Beispiel definiert sind:

- Die Eingabe beginnt mit dem **Messwerteingabe**-Block. Dieser Block wartet auf neu eingehende Messdaten, die einem bestimmten Wert entsprechen, der mit dem Vorlagenparameter **Eingabefragment und Serie** definiert ist. Der Name dieses Parameters ist die Beschriftung, die Sie auf dem Eingabeblock sehen können.
- Die Ausgabe des **Messwerteingabe**-Blocks wird dann an den **Fehlende Daten**-Block übergeben. Der **Fehlende Daten**-Block löst eine Ausgabe aus, wenn innerhalb der mit dem Vorlagenparameter **Dauer (Sekunden)** definierten Zeit keine Eingabe empfangen wird.
- Die Ausgabe des **Fehlende Daten**-Blocks wird als Auslöser für den **Alarm erzeugen**-Eingangsport des **Alarmausgabe**-Blocks verwendet. Der Name des Vorlagenparameters **Alarmtyp** ist die Beschriftung, die Sie auf dem Ausgabeblock sehen können.
- Die Ausgabe des **Messwerteingabe**-Blocks wird auch als Eingabe an den **Objekt**-Eingangsport des **Textersetzung**-Blocks übergeben, zusammen mit der Eingabe aus dem **Managed-Object-Eingabe**-Block, die an den **Quelle**-Eingangsport des **Textersetzung**-Block übergeben wird. Der Name des Vorlagenparameters **Gerät oder Gerätegruppe** ist die Beschriftung, die Sie auf dem Eingabeblock sehen können.
- Der **Textersetzung**-Block unterstützt das Ersetzen von Platzhaltern. Wenn der Eingabetext zum Beispiel „Fehlende Messdaten vom Typ: #{type}“ lautet, dann wird der Wert `#{type}` durch den tatsächlichen `type` der Messung ersetzt. Weitere Informationen finden Sie unter „Textersetzung“ auf Seite 230.
- Der **Textersetzung**-Block ist so konfiguriert, dass er den Vorlagenparameter **Alarmtext** als Benutzereingabe verwendet. Der Block nimmt alle erforderlichen Ersetzungen vor und sendet

die Zeichenfolge mit den Ersetzungen von seinem **Ausgabe**-Ausgangsport an den **Nachricht**-Eingangsport des **Alarmausgabe**-Blocks.

- Der **Alarmausgabe**-Block erfordert die Konfiguration der Vorlagenparameter **Alarmtyp** und **Alarmschweregrad** und erzeugt einen Alarm, wenn er durch den **Fehlende Daten**-Block ausgelöst wird.

Schritt 1: Erstellen Sie ein neues Modell aus einem Beispiel

Im Modellmanager sind auf der Registerkarte **Beispiele** alle Beispielmodelle aufgelistet, die mit Analytics Builder zur Verfügung gestellt werden. Sie können ein Beispiel anzeigen, indem Sie einfach auf seine Karte klicken, aber Sie können es nicht bearbeiten oder aktivieren. Sie können die Beispiele jedoch als Grundlage für die Entwicklung eigener Modelle verwenden, indem Sie ein Modell aus einem Beispiel erstellen.

1. Gehen Sie im Modellmanager zur Registerkarte **Beispiele**.
2. Klicken Sie auf das Aktionsmenü des Beispiels **Bei fehlenden Messdaten Alarm erzeugen** und klicken Sie dann auf **Modell aus Beispiel erstellen**.

   Das neue Modell wird sofort im Modelleditor angezeigt. Es hat denselben Namen, dieselbe Beschreibung und dieselben Tags wie das Beispiel.
3. Wenn Sie das Modell umbenennen möchten, klicken Sie links in der Symbolleiste auf den Modellnamen. Im daraufhin angezeigten Dialogfeld "Modellkonfiguration" können Sie dann einen neuen Namen eingeben.
4. Klicken Sie in der Symbolleiste des Modelleditors auf .
5. Klicken Sie in der Symbolleiste des Modelleditors auf um den Modelleditor zu verlassen und zum Modellmanager zurückzukehren.

   Anmerkung:

   Denken Sie daran, dass auf der Registerkarte **Modelle** des Modellmanagers nur die gespeicherten Modelle angezeigt werden

Schritt 2: Erstellen Sie eine neue Instanz des Modells

Das Beispielmodell benutzt Vorlagenparameter. Wenn Sie also das Beispiel in ein Modell umwandeln, dann erstellen Sie ein sogenanntes Vorlagenmodell. Sie können ein Vorlagenmodell nicht direkt im Modellmanager aktivieren. Stattdessen müssen Sie mindestens eine Instanz des Modells erstellen, und Sie können diese Instanz dann mit dem Instanzeditor aktivieren.

1. Auf der Registerkarte **Modelle** des Modellmanagers: lokalisieren Sie die Karte für das neu erstellte Modell.
2. Klicken Sie auf der Karte auf **0 Instanzen** um den Instanzeditor aufzurufen.
3. Klicken Sie auf **Neue Instanz** um die erste Instanz Ihres neuen Modells zu erstellen.

Schritt 3: Geben Sie die Werte für die Vorlagenparameter ein

Im Instanzeditor wird eine Zeile für jede Instanz angezeigt. Es gibt eine Spalte für jeden im Vorlagenmodell definierten Vorlagenparameter, wobei der Name des Vorlagenparameters die Spaltenüberschrift ist. Solange eine Instanz nicht aktiv ist, können Sie die Werte für diese Instanz ändern.

Benutzen Sie die horizontale Bildlaufleiste, wenn nicht alle Vorlagenparameter (Spalten) auf dem Bildschirm angezeigt werden.

1. Klicken Sie auf das Feld unter der Spaltenüberschrift **Gerät oder Gerätegruppe**. Wählen Sie im daraufhin erscheinenden Dialogfeld das Gerät oder die Gerätegruppe aus, die Sie für diese Instanz verwenden möchten, und klicken Sie auf **Verwenden**.

2. Geben Sie im Textfeld unter der Spaltenüberschrift **Eingabefragment und Serie** die Details der zu überwachenden Messwerteingabe an, und zwar im folgenden Format:

   ```
   <wertFragmentname>.<wertSerienname>
   ```

   Beispiel: Wenn das Fragment `c8y_Gyroscope` und die Serie `gyroscopeX` ist, dann müssen Sie Folgendes eingeben:

   ```
   c8y_Gyroscope.gyroscopeX
   ```

   Tipp:

   Wenn Sie herausfinden möchten, welche Fragmente und Serien für Ihr Gerät verfügbar sind, ohne die vordefinierten Vorlagenparameter des **Messwerteingabe**-Blocks zu ändern, gehen Sie zurück zum Modelleditor und ziehen Sie den Eingabeblock für Ihr Gerät aus der Palette auf die Arbeitsfläche. Öffnen Sie dann das Dropdown-Listenfeld **Fragment und Serie**. Hier werden alle Werte aufgelistet, die Sie verwenden können. Anstelle des `=>`, das Sie im Dropdown-Listenfeld sehen können, müssen Sie in diesem Fall jedoch einen Punkt (.) verwenden. Vergessen Sie nicht, diesen Block wieder zu entfernen, nachdem Sie sich für einen Wert entschieden haben.

3. Die Felder unter den Spaltenüberschriften **Dauer (Sekunden)**, **Alarmtyp**, **Alarmtext** und **Alarmschweregrad** enthalten bereits Standardwerte (siehe auch die Beschreibung der Blöcke oben). Passen Sie diese Werte an Ihre Anforderungen an. Zum Beispiel: ändern Sie die Dauer auf 30 Sekunden, benennen Sie den Alarmtyp in „MeinAlarmtyp“ um, behalten Sie den vordefinierten Alarmtext bei, und ändern Sie den Alarmschweregrad auf **Weniger wichtig**.

4. Klicken Sie in der Symbolleiste des Instanzeditors auf **Speichern**.

Schritt 4: Aktivieren Sie die Instanz

Sie werden nun Ihre neue Instanz im Produktionsmodus aktivieren. Dadurch wird die Instanz eingesetzt und die Messungen von Ihrem Gerät werden verarbeitet.

1. Klicken Sie in der **Modus**-Spalte des Instanzeditors auf das Dropdown-Listenfeld für Ihre Instanz und wählen Sie **Produktion**.
2. Klicken Sie in der **Status**-Spalte des Instanzeditors auf die Umschaltfläche **Inaktiv**, um den Status in **Aktiv** zu ändern

Schritt 5: Senden Sie Daten von Ihrem Gerät

Senden Sie nach der Aktivierung der Instanz die Daten von Ihrem Gerät. Die Instanz beginnt mit der Überwachung des Geräts, sobald die Messdaten eintreffen, und erzeugt einen Alarm, wenn innerhalb der konfigurierten Dauer keine Daten empfangen werden.

Für unseren Beispielfall mit den Gyroskop-Messungen von einem Smartphone sollte es ausreichen, wenn Sie einfach das Smartphone-Display ausschalten, während die Cumulocity IoT Sensor App noch läuft.

Schritt 6: Gehen Sie zum Device Management und zeigen Sie die Alarme an

Um die von Ihrer aktiven Instanz gesendeten Alarme anzuzeigen, müssen Sie zur Device Management-Anwendung von Cumulocity IoT wechseln. Ausführliche Informationen finden Sie unter Device Management in der Cumulocity IoT-Dokumentation.

1. Klicken Sie auf die Schaltfläche für den Application Switcher ( ) und dann auf **Device Management**.
2. Klicken Sie im Navigator auf der linken Seiten auf **Geräte** und dann auf **Alle Geräte**.
3. Lokalisieren Sie Ihr Gerät und klicken Sie auf dessen Name um die Gerätedetails anzuzeigen.
4. Klicken Sie links auf **Alarme**.
5. Überprüfen Sie auf der daraufhin erscheinenden Seite die Alarme, die von Ihrem Gerät gesendet werden. Wenn Sie Ihre Instanz wie oben beschrieben editiert haben, dann sollten Sie Ihren Alarm nach 30 Sekunden unter WENIGER WICHTIG sehen, und zwar mit dem folgenden Text: „Fehlende Messwerte vom Typ: c8y_Gyroscope“. Weitere Informationen zum **Alarme**-Tab finden Sie in der Cumulocity IoT-Dokumentation unter Verwenden von Alarmen.

3 Modelle verstehen

- Modelle 40
- Vorlagenmodellinstanzen 41
- Blöcke 41
- Verbindungen 45
- Anwendungsbeispiel 45

Modelle

Ein Modell ist ein Container für ein Netzwerk von Blöcken, die über Verbindungen miteinander verknüpft sind.

Das Verhalten eines Blocks innerhalb eines Modells ist nicht von anderen Blöcken abhängig. In einem Modell können mehrere Instanzen desselben Blocks enthalten sein, wobei sich jede Instanz - in Abhängigkeit von den konfigurierten Parametern oder den mit dem Block verbundenen Eingängen (Ports) - anders verhalten kann.

Sie können zwei verschiedene Arten von Modellen erstellen: Modelle ohne Vorlagenparameter und Modelle mit Vorlagenparametern.

Modelle ohne Vorlagenparameter

Alle Blöcke in dem Modell verwenden definierte Eingabegeräte oder Gerätegruppen und enthalten definierte Parameterwerte. Ein solches Modell kann sofort im Modellmanager aktiviert werden.

Modelle mit Vorlagenparametern

Ein Modell, in dem ein oder mehrere Vorlagenparameter definiert sind, wird als „Vorlagenmodell“ bezeichnet. Vorlagenparameter können an eine beliebige Anzahl von Blockparametern gebunden werden. Die Voraussetzung hierfür ist, dass der Typ des Blockparameters dem Typ des Vorlagenparameters entspricht.

Sie können zum Beispiel einen Vorlagenparameter für den Gerätenamen und einen anderen für den Schwellenwert definieren. Diese Vorlagenparameter können später individuell in den einzelnen Instanzen des Modells gesetzt werden. Ein Vorlagenparameter kann zum Beispiel ein Gerät benennen, welches dann bei verschiedenen Ein- und Ausgabeblöcken verwendet werden kann. Oder eine Instanz kann das Gerät ABC mit einem Schwellenwert von 100 benutzen, während eine andere Instanz das Gerät XYZ mit einem Schwellenwert von 200 benutzt. Modelle mit Vorlagenparametern werden nicht direkt im Modellmanager aktiviert. Sie müssen mindestens eine Instanz des Modells erstellen, und danach können Sie jede Instanz separat im Instanzeditor aktivieren.

Der Geltungsbereich der Vorlagenparameter ist lokal für das Modell, in dem sie definiert sind. Mit anderen Worten: Vorlagenparameter, die in einem Modell definiert sind, können nicht in einem anderen Modell verwendet werden, das auf demselben Mandanten oder einem Untermandanten bereitgestellt wird. Die Namen der Vorlagenparameter müssen innerhalb des Modell, in dem sie definiert sind, eindeutig sein.

Es gibt zwei relevante Rollen für diesen Modelltyp. Diese Rollen können von derselben Person oder von verschiedenen Personen ausgeübt werden.

- **Modellautor**

  Der Modellautor erstellt das Modell und definiert alle Blöcke, Parameter und Verbindungen in dem Modell. Hierbei erstellt der Modellautor auch die Vorlagenparameter und bindet sie in ausgewählten Blöcken an die entsprechenden Parameter.

- **Instanzenverwalter**

Der Instanzenverwalter erstellt die Instanzen des Modells und definiert die Werte für die Vorlagenparameter, die in den einzelnen Instanzen verwendet werden sollen.

Der Modellautor hat die folgenden Möglichkeiten, einen Vorlagenparameter zu definieren:

- Der Vorlagenparameter kann einen Standardwert haben, der im Instanzeditor als Standardwert zu sehen ist. Der Instanzenverwalter kann den Standardwert dann beibehalten oder in einen anderen Wert abändern.
- Der Vorlagenparameter kann optional sein. Der Instanzenverwalter hat dann die Möglichkeit, einen Wert anzugeben oder ihn leer zu lassen.
- Der Vorlagenparameter kann erforderlich sein. Der Instanzenverwalter muss dann einen Wert angeben. Ein erforderlicher Wert ist ein Wert, der nicht optional ist und der keinen Standardwert hat.

Vorlagenmodellinstanzen

Vorlagenmodellinstanzen enthalten die Werte, die in den Modellen mit Vorlagenparametern verwendet werden sollen (siehe auch „Modelle“ auf Seite 40).

Beispiel: zwei Geräte führen ähnliche Prüfungen der Gerätedaten durch, aber sie verwenden hierfür unterschiedliche Schwellenwerte. In diesem Fall würden Sie für jedes Gerät eine separate Instanz konfigurieren und in jeder Instanz das zu verwendende Gerät und den Schwellenwert angeben.

Jede Instanz kann unabhängig von den anderen Instanzen aktiviert und deaktiviert werden, und jede Instanz kann in einem anderen Modus ausgeführt werden.

Blöcke

Die Blöcke sind die wesentlichen Verarbeitungseinheiten eines Modells. Jeder Block hat bestimmte vordefinierte Funktionen und verarbeitet die Daten entsprechend. Ein Block kann ein Reihe von Parametern haben sowie eine Reihe von Ein- und Ausgangsports.

Die Palette im Modelleditor bietet die folgenden Blocktypen zur Auswahl an:

- Eingabeblöcke, die Daten von externen Quellen empfangen. Ein Eingabeblock steht normalerweise für ein im Cumulocity IoT-Inventar registriertes Gerät oder eine Gerätegruppe. Siehe auch „Eingabeblöcke“ auf Seite 42.
- Ausgabeblöcke, die Daten an externe Quellen senden. Ein Ausgabeblock steht normalerweise für ein im Cumulocity IoT-Inventar registriertes Gerät. Es gibt aber auch Ausgabeblöcke für das Senden von E-Mail oder SMS an angegebene Empfänger. Siehe auch „Ausgabeblöcke“ auf Seite 42.
- Verarbeitungsblöcke, die Daten von den Eingabeblöcken empfangen und die resultierenden Daten an die Ausgabeblöcke senden. Siehe auch „Verarbeitungsblöcke“ auf Seite 43.

Anmerkung:

Detaillierte Informationen zu jedem Block finden Sie in der Blockreferenz. Links zu den Beschreibungen aller Blöcke finden Sie unter „Übersicht aller Blöcke“ auf Seite 164.

Ein Block kann über seine Eingangsports Daten von einem anderen Block empfangen. Ein Block kann über seine Ausgangsports Daten an einen anderen Block senden. Die verschiedenen Blöcke haben eine unterschiedliche Anzahl von Ein- und Ausgangsports, und einige Blöcke haben nur Eingangsports oder nur Ausgangsports. Bei den meisten Blöcken ist es nicht erforderlich, alle Ein- oder Ausgangsports zu verbinden.

Ein Block kann konfigurierbare Parameter haben, die das Verhalten des Blocks definieren. Diese Parameter sind entweder optional oder erforderlich, je nach den Anforderungen des Blocks. Ein Parameter kann mit einem Wert oder einem Vorlagenparameter konfiguriert werden.

Wenn Sie denselben Block mehrmals verwenden, können Sie unterschiedliche Werte für denselben Parameter angeben. Der **Schwelle**-Block hat zum Beispiel einen konfigurierbaren Parameter mit dem Namen **Schwellenwert**. Wenn Sie zwei Instanzen des **Schwelle**-Blocks verwenden und diesen Parameter bei jedem Block anders konfigurieren, dann melden die Blöcke unterschiedliche Übertretungen des Schwellenwerts.

Anmerkung:

Zwei Ausgangsports können nicht an denselben Eingangsport angeschlossen werden. Aber ein Ausgangsport kann an mehrere Eingangsports angeschlossen werden.

Eingabeblöcke

Ein Eingabeblock ist ein spezieller Blocktyp, der Daten von einer externen Quelle empfängt. Der Block verwandelt die Daten in ein für die Verbindung verständliches Format und überträgt sie an die verbundenen Blöcke. Wenn ein Eingabeblock zum Beispiel ein `Measurement`-Ereignis von Cumulocity IoT empfängt, dann extrahiert der Block die erforderlichen Informationen aus dem Ereignis und überträgt die Informationen anschließend zur weiteren Verarbeitung an die verbundenen Blöcke.

Modelle können Daten von mehreren Geräten verarbeiten und dabei (bei Verwendung von mehreren Kernen) skalieren. Ausführliche Informationen finden Sie unter „Modellausführung für verschiedene Geräte“ auf Seite 134.

Darüber hinaus unterstützt Analytics Builder Eingabegeräte, die als „Broadcast-Geräte“ bezeichnet werden. Die Signale von diesen Geräten stehen allen Modellen und Geräten zur Verfügung. Ausführliche Informationen finden Sie unter „Broadcast-Geräte“ auf Seite 136.

Ausgabeblöcke

Ein Ausgabeblock ist ein spezieller Blocktyp, der Daten von einem verbundenen Verarbeitungsblock empfängt. Der Ausgabeblock verwandelt die Daten in ein für eine externe Quelle verständliches Format und überträgt sie an die externe Quelle. Wenn ein Ausgabeblock zum Beispiel Daten von einem verbundenen Verarbeitungsblock empfängt, dann packt der Ausgabeblock die Daten in ein `Operation`-Objekt und sendet die Operation anschließend an Cumulocity IoT.

Der Block mit dem Namen **Trigger-Gerät** ist ein spezieller Ausgabeblock, mit dem die Ausgabe an das Gerät zurückgeschickt werden kann, das die Ausgabe ausgelöst hat. Modelle können Daten von mehreren Geräten verarbeiten und dabei (bei Verwendung von mehreren Kernen) skalieren. Ausführliche Informationen finden Sie unter „Modellausführung für verschiedene Geräte“ auf Seite 134.

Andere Ausgabeblöcke sind **E-Mail senden** und **SMS senden** zum Versenden von E-Mails und Textnachrichten. Diese Blöcke erfordern, dass die Mandantenumgebung korrekt konfiguriert wurde, damit die E-Mails und Textnachrichten ausgeliefert werden können (siehe auch Providing SMS provider credentials in der Cumulocity IoT-Dokumentation (hiervon gibt es auch eine deutsche Version). Im Gegensatz zu den anderen Blöcken sind diese nicht mit Geräten innerhalb der Cumulocity IoT-Platftorm verbunden.

Verarbeitungsblöcke

Es gibt verschiedene Arten von Verarbeitungsblöcken. In der Palette des Modelleditors werden sie je nach Funktionalität in verschiedene Kategorien eingeteilt.

Diese Kategorie	enthält Blöcke die
Logik	logische Operationen mit den Daten durchführen. Blöcke wie **UND** und **ODER** gehören zu dieser Kategorie.
Berechnung	mathematische Operationen mit den Daten durchführen. Blöcke wie **Ausdruck**, **Delta**, **Differenz**, **Richtungserkennung** und **Schwelle** gehören zu dieser Kategorie.
Aggregat	Aggregation der Daten über ein Fenster mit Werten durchführen. Blöcke wie **Durchschnitt (Mittelwert)** und **Integral** gehören zu dieser Kategorie.
Fluss-Manipulation	den Datenfluss manipulieren. Blöcke wie **Gate**, **Latch-Werte**, **Pulse** und **Zeitverzögerung** gehören zu dieser Kategorie.
Werkzeug	verschiedene Hilfsfunktionen anbieten. Blöcke wie **Fehlende Daten** und **Umschalten** gehören zu dieser Kategorie.

Beispiel eines Verarbeitungsblocks - der Schwelle-Block

Das folgende Beispiel zeigt, wie ein Block und der Blockparametereditor im Modelleditor aussehen. Es zeigt den **Schwelle**-Block, mit dem Sie feststellen können, ob der Eingabewert den Schwellenwert durchbricht oder überschreitet.

Die Parameter sind:

- **Schwellenwert**. `float`-Typ. Dieser Wert wird mit dem Eingabewert verglichen.
- **Richtung**. Die Richtung, in der verglichen soll: Ob der Eingabewert oberhalb oder unterhalb des definierten Schwellenwerts liegt oder ob er die Schwelle überschreitet.

Die Eingangsports sind:

- **Wert**. `float`-Typ. Der Eingabewert, der mit dem definierten Schwellenwert verglichen werden soll.
- **Zurücksetzen**. `pulse`-Typ. Wenn ein Pulse-Signal empfangen wird, dann wird der Zustand des Blocks zurückgesetzt, so dass die zuvor empfangenen Eingabewerte nicht mehr verwendet werden.

Die Ausgangsports sind:

- **Schwelle durchbrochen**. `boolean`-Typ. Wird auf `true` gesetzt, wenn die Schwelle durchbrochen wurde. Das heißt, der Eingabewert liegt nicht im Bereich des definierten Schwellenwerts.
- **Innerhalb Schwelle**. `boolean`-Typ. Wird auf `true` gesetzt, wenn die Schwelle nicht durchbrochen wurde. Das heißt, der Eingabewert liegt im Bereich des definierten Schwellenwerts.

- **Schwelle überschritten**. `pulse`-Typ. Sendet ein Pulse-Signal, wenn der Eingabewert die Schwelle überschreitet (und somit von einer Seite der Schwelle zur anderen geht).

Eigene Blöcke erstellen

Mit dem Analytics Builder Block SDK können Sie Ihre eigenen, benutzerdefinierten Blöcke schreiben, testen, verpacken und nach Analytics Builder hochladen.

Das Block SDK finden Sie in GitHub unter https://github.com/SoftwareAG/apama-analytics-builder-block-sdk. Ausführliche Information zum Block SDK (auf Englisch) finden Sie in der Dokumentation in GitHub.

Sie schreiben die benutzerdefinierten Blöcke mit der Event Processing Language (EPL) von Apama. Sobald Sie einen Block geschrieben haben, können Sie ihn in eine *Extension* packen und hochladen. Eine Beispielbefehlszeile zum Erstellen und Hochladen einer Extension lautet:

```
analytics_builder build extension --input pfad --cumulocity_url $C8Y_URL --username
$C8Y_USERNAME --password $C8Y_PASSWORD --name customBlocks --restart
```

Um eine Extension hochzuladen, benötigt der im Argument `--username` angegebene Benutzer - zusätzlich zu den in „Voraussetzungen“ auf Seite 28 aufgeführten Berechtigungen - die folgende Berechtigung in Cumulocity IoT: ERSTELLEN für "Stammdaten" (englisch: Inventory).

Der Apama-ctrl-Microservice wird nach Ausführung des obigen Befehls neu gestartet. Um einen Neustart auszuführen, benötigt der Benutzer die folgende Berechtigung in Cumulocity IoT: ADMIN für "Regeln" (englisch: CEP management).

Verbindungen

Ein Block wird mit Hilfe von Verbindungen mit einem anderen Block verknüpft. Die gesamte Datenübertragung zwischen dem Ausgangsport eines Blocks und dem Eingangsport eines anderen Blocks erfolgt über diese Verbindungen. Alle Verbindungen müssen zwischen kompatiblen Typen hergestellt werden. Ausführliche Informationen finden Sie unter „Verbindungen und Blöcke“ auf Seite 99.

Anmerkung:

Im Blocknetzwerk eines Modells ist nicht möglich, Kreisläufe mit den Verbindungen zu erzeugen. Siehe auch „Verbindungseinschränkungen“ auf Seite 109.

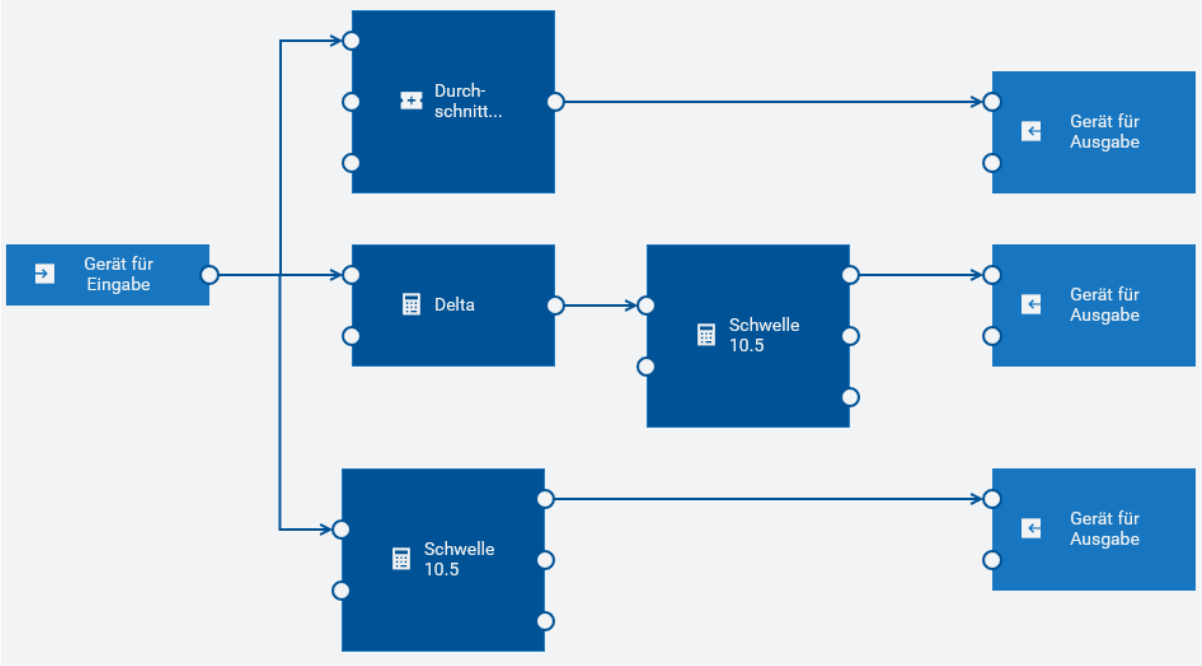
Anwendungsbeispiel

Stellen Sie sich vor, dass Sie Sensordaten in Echtzeit erhalten und diese Daten analysieren möchten. Nehmen wir zur Vereinfachung an, dass es nur einen Sensor gibt und dass Sie sich für Folgendes interessieren:

- Sie möchten den Durchschnittswert der Sensorwerte über einen bestimmten Zeitraum ermitteln.
- Sie möchten plötzliche Änderungen der Sensorwerte anhand eines definierten Schwellenwerts erkennen.

- Sie möchten sicherstellen, dass die Sensorwerte innerhalb eines bestimmten Bereichs liegen und dass eine Warnung ausgegeben wird, wenn die Messwerte diesen Bereich überschreiten. Zum Beispiel: Sie erhalten Druckwerte und möchten sicherstellen, dass der maximale Druck nicht über den Bereich hinausgeht, der von dem Gerät bewältigt werden kann.

Das Modell für dieses Beispiel enthält die folgenden Blöcke:



- Ein Eingabeblock, der **Gerät für Eingabe** als Gerätename anzeigt

  Die eingehenden Daten sind in Echtzeit und kontinuierlich. Der Eingabeblock empfängt die Daten vom Sensor. Die Daten werden an die **Durchschnitt (Mittelwert)**-, **Delta**- und **Schwelle**-Blöcke übergeben. Die Eingangsports dieser Blöcke sind mit dem Ausgangsport des Eingangsblocks verbunden.

- Ein **Durchschnitt (Mittelwert)**-Block

  Dieser Block ermittelt den Durchschnitt (oder Mittelwert) der Messwerte, die er über einen bestimmten Zeitraum empfängt, und leitet dies an den verbundenen Ausgangsblock weiter.

- Ein **Delta**-Block

  Dieser Block berechnet die Differenz zwischen aufeinanderfolgenden Eingabewerten und leitet den berechneten Wert an den verbundenen **Schwelle**-Block weiter.

- Zwei verschiedene Instanzen des **Schwelle**-Blocks

  Ein **Schwelle**-Block vergleicht den Eingabewert mit dem definierten Schwellenwert, um festzustellen, ob der Eingabewert den Schwellenwert durchbricht oder nicht.

  Die erste Instanz ist mit dem **Delta**-Block verbunden und meldet einen Verstoß, wenn der Deltawert den Schwellenwert durchbricht.

Die zweite Instanz ist mit dem Eingabeblock verbunden und meldet einen Verstoß, wenn der Eingabewert nicht innerhalb des Schwellenwerts liegt.

- Drei Instanzen eines Ausgabeblocks, die **Gerät für Ausgabe** als Gerätenamen anzeigen

  Die erste Instanz sendet den Durchschnitt der Sensorwerte.

  Die zweite Instanz generiert eine Ausgabe, wenn sich die Werte von aufeinanderfolgenden Sensorablesungen um mehr als den konfigurierten Schwellenwert ändern.

  Die dritte Instanz erzeugt eine Ausgabe, wenn der Sensorwert den konfigurierten Schwellenwert überschreitet.

4 Mit dem Modellmanager arbeiten

- Die Benutzeroberfläche des Modellmanagers 50
- Modelle und Beispiele filtern 52
- Ein neues Modell erstellen 54
- Ein bestehendes Modell bearbeiten 55
- Die Instanzen eines Modells bearbeiten 56
- Ein Modell aktivieren 56
- Ein Modell deaktivieren 58
- Ein Modell duplizieren 58
- Ein Modell exportieren 59
- Ein Modell importieren 59
- Ein Modell entfernen 59
- Alle Modelle neu laden 60
- Ein Beispiel anzeigen 60
- Ein Modell aus einem Beispiel erstellen 60

Die Benutzeroberfläche des Modellmanagers

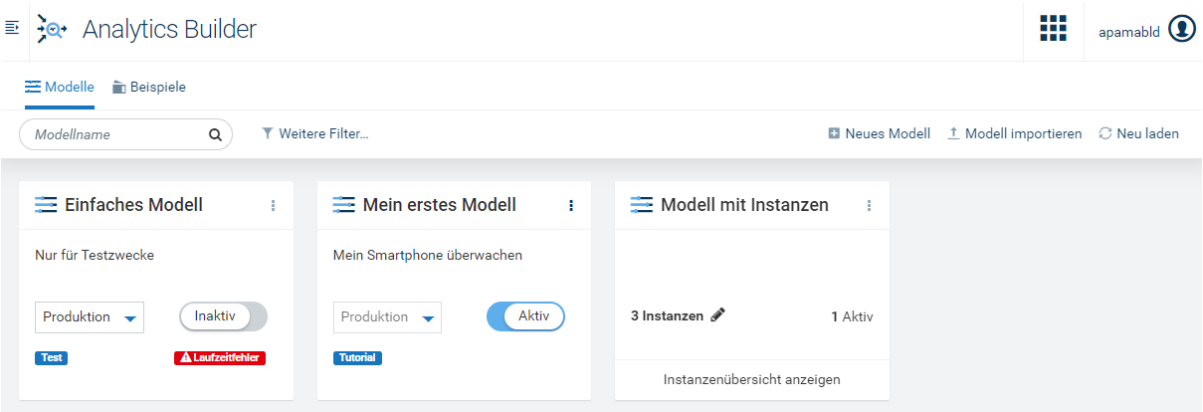
Der Modellmanager enthält zwei Registerkarten: die Registerkarte **Modelle**, auf der alle zur Zeit definierten Modelle angezeigt werden, und die Registerkarte **Beispiele**, auf der Beispielmodelle angezeigt werden, die Ihnen den Einstieg in die Erstellung Ihrer eigenen Modelle erleichtern sollen.

Anmerkung:

Detaillierte Informationen zu den Elementen in der oberen Leiste (Application Switcher und Benutzer-Schaltfläche) finden Sie in der Cumulocity IoT-Dokumentation.

Die Registerkarte "Modelle"

Die Registerkarte **Modelle** listet alle verfügbaren Analysemodelle in der aktuellen Cumulocity IoT-Umgebung als Karten auf. Von hier aus erstellen Sie neue Modelle und verwalten die vorhandenen Modelle.



Um ein Modell zu bearbeiten, klicken Sie einfach auf die Karte, die für das Modell angezeigt wird (siehe auch „Ein bestehendes Modell bearbeiten“ auf Seite 55). Wenn Sie ein neues Modell erstellen oder ein vorhandenes Modell bearbeiten, wird der Modelleditor aufgerufen, in dem Sie die Blöcke und Verbindungen definieren, aus denen ein Modell besteht. Ausführliche Informationen finden Sie unter „Mit dem Modelleditor arbeiten“ auf Seite 63.

Es gibt zwei Arten von Modellen, und die Karten für diese Modelle sehen unterschiedlich aus:

- Wenn eine Karte den Modus (zum Beispiel **Entwurf** oder **Produktion**) und den Status (**Aktiv** oder **Inaktiv**) anzeigt, dann handelt es sich um ein Modell ohne Vorlagenparameter. Ein solches Modell kann sofort im Modellmanager aktiviert werden. Weitere Informationen finden Sie unter „Ein Modell aktivieren“ auf Seite 56.

  Wenn Laufzeitfehler auf der Karte eines aktiven Modells angezeigt wird, dann verarbeitet dieses Modell keine Ereignisse mehr. Klicken Sie das Fehlersymbol an, um Informationen darüber anzuzeigen, was schief gelaufen ist.

- Wenn ein Modell Vorlagenparameter hat, dann dient es als Vorlage. In diesem Fall wird die Anzahl der definierten und aktiven Instanzen auf der Karte angezeigt. Ein Vorlagenmodell

wird nicht direkt im Modellmanager aktiviert. Stattdessen müssen Sie den Instanzeditor verwenden und eine Reihe von Instanzen erstellen, wobei jede Instanz bestimmte Werte für die Vorlagenparameter zur Verfügung stellt. Jede Instanz hat einen Modus und kann im Instanzeditor aktiviert und deaktiviert werden (auf dieselbe Art und Weise wie Modelle ohne Vorlagenparameter im Modellmanager aktiviert und deaktiviert werden).

Um die Instanzen zu bearbeiten, können Sie einfach auf die Gesamtzahl der Instanzen klicken (siehe auch „Die Instanzen eines Modells bearbeiten“ auf Seite 56). Dies ruft den Instanzeditor auf. Ausführliche Informationen finden Sie unter „Mit dem Instanzeditor arbeiten“ auf Seite 91.

Sie können die Karte für ein Vorlagenmodell drehen, um weitere Details anzuzeigen. Klicken Sie hierzu auf **Instanzenübersicht anzeigen**. Sie können dann sehen wie viele Instanzen in einem bestimmten Modus ausgeführt werden.

Wenn ein Fehlersymbol (zum Beispiel

) auf der Karte eines Vorlagenmodells angezeigt wird, dann verarbeitet mindestens eine der Instanzen keine Ereignisse mehr. Klicken Sie das Fehlersymbol an, um Informationen darüber anzuzeigen, was schief gelaufen ist.

Solange ein Modell keine Vorlagenparameter hat, gibt es null Instanzen und die Karte zeigt die Steuerelemente zum Auswählen und Aktivieren eines Modus.

Jede Karte, die für ein Modell angezeigt wird, hat ein Aktionsmenü (die drei vertikalen Punkte, die oben rechts auf der Karte angezeigt werden), welches Befehle zur Verwaltung des Modells enthält (zum Beispiel zum Exportieren oder Entfernen des Modells).

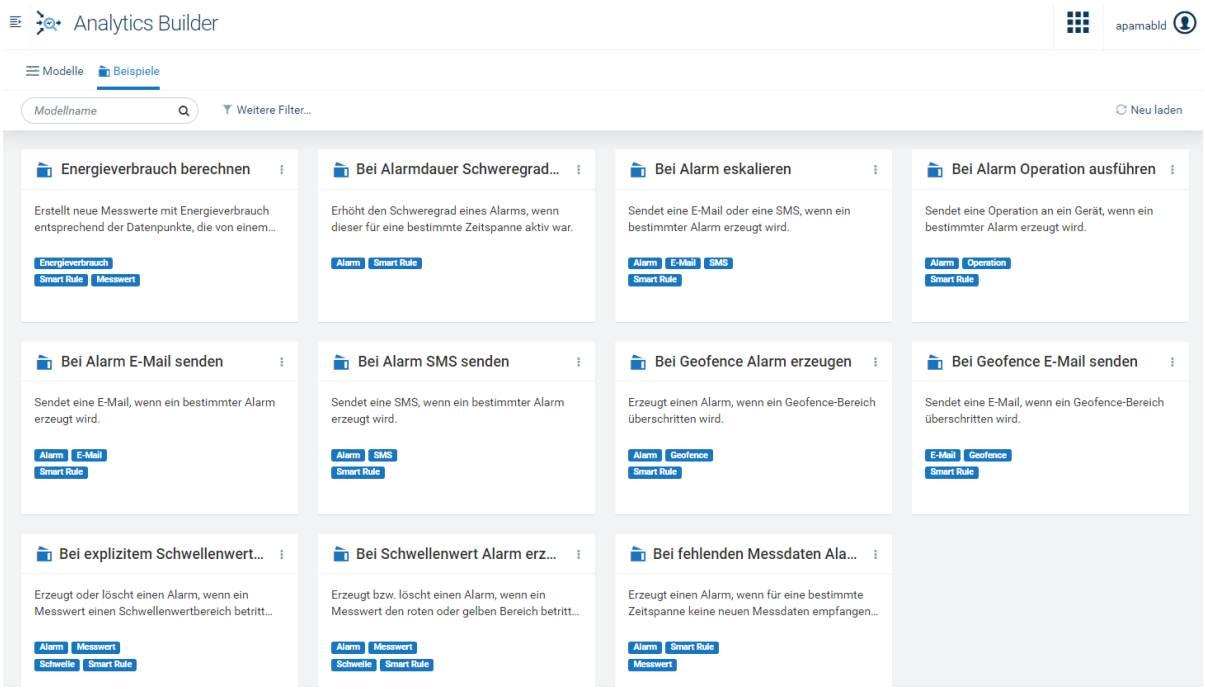
Wenn für ein Modell eine Beschreibung oder Tags definiert wurden, dann wird dies auf der Karte für dieses Modell angezeigt. Wenn Sie den Namen, die Beschreibung oder die Tags eines vorhandenen Modells ändern möchten, müssen Sie dies im Modelleditor tun. Siehe „Name, Beschreibung und Tags eines Modells ändern“ auf Seite 66.

Wenn Sie eine lange Liste von Karten haben, dann können Sie das gewünschte Modell leicht finden, indem Sie den Namen des Modells im Suchfeld **Modellname** eingeben. Sie können auch nur einen Teil des Modellnamens eingeben. Sie können zum Beispiel das Wort "test" eingeben, um alle Modelle zu finden, die dieses Wort im Namen enthalten. Die von Ihnen eingegebenen Zeichen können im Modellnamen an einer beliebigen Position stehen. Diese Suchkriterien unterscheiden nicht zwischen Groß- und Kleinschreibung. Wenn Ihre Suchkriterien gerade angewendet werden, dann wird **Suche löschen** neben dem Suchfeld angezeigt. Wenn Sie dies anklicken, dann werden die Suchkriterien gelöscht und alle verfügbaren Karten werden wieder angezeigt.

Sie können die Anzahl der angezeigten Karten auch mit Hilfe eines Filters reduzieren. Ausführliche Informationen hierzu finden Sie unter „Modelle und Beispiele filtern“ auf Seite 52.

Die Registerkarte "Beispiele"

Auf der Registerkarte **Beispiele** sind alle Beispielmodelle als Karten aufgelistet, die mit Analytics Builder zur Verfügung gestellt werden.



Wenn der Name eines Beispiels oder dessen Beschreibung nicht vollständig auf der Karte angezeigt wird, dann können Sie die Maus über den Namen oder die Beschreibung bewegen, um den vollständigen Namen bzw. die vollständige Beschreibung in einem Tooltip anzuzeigen.

Sie können die Beispiele anzeigen, aber nicht ändern oder aktivieren. Um ein Beispiel anzuzeigen, können Sie einfach auf die Karte klicken, die für das Beispiel angezeigt wird. Der Modelleditor wird dann im schreibgeschützten Modus aufgerufen. Weitere Informationen finden Sie unter „Ein Beispiel anzeigen“ auf Seite 60.

Wenn Sie ein Beispiel als Grundlage für die weitere Entwicklung verwenden möchten, dann können Sie aus dem Beispiel ein Modell erstellen. Anschließend können Sie das neue Modell Ihren Anforderungen entsprechend ändern und aktivieren. Weitere Informationen finden Sie unter „Ein Modell aus einem Beispiel erstellen“ auf Seite 60.

Sie können ein Beispiel leicht finden, indem Sie den Namen des Beispiels oder einen Teil des Namens im Suchfeld **Modellname** eingeben (zum Beispiel, "Geofence" oder "E-Mail"). Sie geben die Suchkriterien genau so ein wie es oben für das Suchfeld **Modellname** auf der Registerkarte **Modelle** beschrieben ist. Sie können die Beispiele auch nach Tags filtern; weitere Informationen finden Sie unter „Modelle und Beispiele filtern“ auf Seite 52.

Modelle und Beispiele filtern

Der Modellmanager bietet mehrere Möglichkeiten, die Anzahl der Karten zu reduzieren, die auf den Registerkarten **Modelle** und **Beispiele** angezeigt werden. So können Sie die benötigten Modelle oder Beispiele schneller finden.

Das Filtern funktioniert auch zusammen mit einem Modell- oder Beispielnamen, der im Suchfeld **Modellname** eingegeben wurde (siehe auch die Beschreibung dieses Suchfelds in „Die Benutzeroberfläche des Modellmanagers“ auf Seite 50).

Modelle oder Beispiele filtern

1. Auf der Registerkarte **Modelle** oder **Beispiele** des Modellmanagers: klicken Sie in der Symbolleiste auf **Weitere Filter**.

2. Wählen Sie im daraufhin angezeigten Dialogfeld einen oder mehrere Filter aus. Bei den Beispielen ist es nur möglich nach Tags zu filtern.

   Auf der Registerkarte **Modelle** können Sie die Modelle nach folgenden Kriterien filtern:

   - **Modus**

     Es ist möglich nur die Modelle anzuzeigen, die sich in einem bestimmten Modus befinden. Wenn Sie zum Beispiel nur die Modelle sehen möchten, die sich im Simulations- und Testmodus befinden, aktivieren Sie die entsprechenden Kontrollkästchen.

   - **Status**

     Es ist möglich nur die Modelle anzuzeigen, die sich im Status "Aktiv" oder "Inaktiv" befinden. Wenn Sie zum Beispiel nur aktive Modelle anzeigen möchten, aktivieren Sie das entsprechende Kontrollkästchen.

   - **Gerät oder Gerätegruppe**

     Es ist möglich nur die Modelle anzuzeigen, die Folgendes verwenden: bestimmte Geräte in ihren Ein- und Ausgabeblöcken, bestimmte Gerätegruppen in ihren Eingabeblöcken, oder sogar bestimmte Assets. Öffnen Sie das Dropdown-Listenfeld **Nach Gerätenamen oder Gerätegruppe filtern**, wählen Sie ein oder mehrere Geräte, Gerätegruppen und/oder Assets aus und klicken Sie auf **Anwenden**.

   - **Datenpunkt**

     Es ist möglich nur die Modelle anzuzeigen, die bestimmte Datenpunkte verwenden (wie zum Beispiel `c8y_TemperatureMeasurement`). Hierzu muss mindestens ein Gerät im Dropdown-Listenfeld **Nach Gerätenamen oder Gerätegruppe filtern** ausgewählt sein. Öffnen Sie das Dropdown-Listenfeld **Nach Datenpunkten filtern**, wählen Sie einen oder mehrere Datenpunkte aus und klicken Sie auf **Anwenden**.

   - **Tags**

     Es ist möglich nur die Modelle anzuzeigen, für die bestimmte Tags im Dialogfeld "Modelkonfiguration" definiert wurden. Dieses Dialogfeld wird angezeigt, wenn Sie ein neues Modell erstellen oder wenn Sie dieses Dialogfeld im Modelleditor aufrufen (siehe auch „Ein neues Modell erstellen“ auf Seite 54 und „Name, Beschreibung und Tags eines Modells ändern“ auf Seite 66). Öffnen Sie das Dropdown-Listenfeld **Nach Tag filtern**, wählen Sie ein oder mehrere Tags aus und klicken Sie auf **Anwenden**.

   Sie können mehrere Filtertypen kombinieren. So können Sie zum Beispiel nur die aktiven Modelle im Produktionsmodus anzeigen, die ein bestimmtes Gerät verwenden.

Auf der Registerkarte **Beispiele** können Sie die Beispiele nur nach Tag filtern. Öffnen Sie das Dropdown-Listenfeld **Nach Tag filtern**, wählen Sie ein oder mehrere Tags aus und klicken Sie auf **Anwenden**.

Alle oben genannten Dropdown-Listenfelder enthalten ein **Filter**-Suchfeld, mit dem Sie die Anzahl der zur Auswahl angebotenen Elemente reduzieren können. Sie können einen Namen oder einen Teil eines Namens eingeben. Geben Sie beispielsweise das Wort "test" ein, um nur die Elemente anzuzeigen, in deren Name dieses Wort enthalten ist. Die von Ihnen eingegebenen Zeichen können an einer beliebigen Position innerhalb des Namens enthalten sein. Diese Filterkriterien unterscheiden nicht zwischen Groß- und Kleinschreibung. Wenn Sie auf das Kontrollkästchen **Alle** klicken, werden alle Elemente ausgewählt, die gerade im Dropdown-Listenfeld angezeigt werden, in Abhängigkeit vom Inhalt des Suchfelds **Filter**.

3. Klicken Sie auf **Filter anwenden**.

   In der Symbolleiste des Modellmanagers werden nun die zur Zeit angewendeten Filtertypen angezeigt. Zum Beispiel:

   Klicken Sie auf **Filter entfernen** in der Symbolleiste, wenn Sie diese Filter entfernen möchten. Oder wenn Sie einen bestimmten Filter entfernen möchten, klicken Sie auf das im Filtersymbol angezeigte ×, oder klicken Sie auf den Filternamen im Symbol und deaktivieren Sie diesen Filter (und weitere Filter, falls erforderlich) im daraufhin angezeigten Dialogfeld. Wenn Sie in diesem Dialogfeld auf **Filter zurücksetzen** klicken, dann werden alle Filter entfernt.

Ein neues Modell erstellen

Wenn Sie ein neues Modell erstellen, wird der Modelleditor aufgerufen. Ausführliche Informationen finden Sie unter „Mit dem Modelleditor arbeiten“ auf Seite 63.

Anmerkung:

Das neue Modell wird nur dann im Modellmanager aufgelistet, wenn Sie das Modell im Modelleditor speichern. Siehe auch „Ein Modell speichern“ auf Seite 66.

Sie können ein neues Modell auch aus einem Beispiel erstellen. Siehe „Ein Modell aus einem Beispiel erstellen“ auf Seite 60.

Ein neues Modell erstellen

1. Auf der Registerkarte **Modelle** des Modellmanagers: klicken Sie in der Symbolleiste auf **Neues Modell**.

2. Geben Sie im anschließend angezeigten Dialogfeld "Modellkonfiguration" einen eindeutigen Modellnamen ein.

Optional können Sie eine Beschreibung für das Modell und ein oder mehrere Tags eingeben.

Tags sind hilfreich, wenn Sie die Modelle im Modellmanager filtern und nur die Modelle anzeigen möchten, für die ein bestimmtes Tag definiert wurde (siehe auch „Modelle und Beispiele filtern“ auf Seite 52). Um ein Tag hinzuzufügen, geben Sie einfach seinen Namen ein und drücken die Eingabe- oder Tabulatortaste. Das Tag wird dann in einem farbigen Rechteck angezeigt. Um ein Tag zu entfernen, klicken Sie auf das im Rechteck angezeigte . Das Dialogfeld verhindert, dass Sie für ein Modell doppelte Tags eingeben; wenn Sie einen solchen Tag-Namen eingeben, wird das doppelte Tag nicht hinzugefügt und das bestehende Tag blinkt einmal.

3. Klicken Sie auf **OK**.

   Der Modelleditor wird angezeigt. Unter „Übersicht der Schritte für das Erstellen eines Modells“ auf Seite 65 finden Sie einen kurzen Überblick darüber, wie Sie dem neuen Modell Blöcke und Verbindungen hinzufügen können.

Ein bestehendes Modell bearbeiten

Sie können jedes Modell bearbeiten (oder anzeigen), das zur Zeit im Modellmanager aufgelistet wird.

Wenn sich ein Modell im Status "Aktiv" befindet, dann wird das Modell durch das Bearbeiten in den Lesemodus versetzt. Dann ist es nur möglich, den Inhalt des Modells im Modelleditor zu betrachten (Sie können sich zum Beispiel die Blockparameter ansehen). Sie können im Modell wie gewohnt navigieren und zoomen, aber Sie können nichts ändern. Die Schaltfläche zum Speichern ist daher im Modelleditor deaktiviert.

Ein Modell bearbeiten

- Auf der Registerkarte **Modelle** des Modellmanagers: klicken Sie einfach auf die Karte, die für das Modell angezeigt wird (jedoch nicht auf die Umschaltfläche zum Ändern des Status oder das Dropdown-Menü zum Ändern des Modus).

  Oder klicken Sie auf das Aktionsmenü des Modells und klicken Sie dann auf **Bearbeiten**.

  Wenn sich das Modell im Status "Aktiv" befindet, erscheint ein Dialogfeld und Sie werden informiert, dass Sie sich das Modell nur ansehen können. Wenn Sie auf **Weiter** klicken, wird der Modelleditor angezeigt und Sie können sich das Modell ansehen, aber nicht ändern.

  Weitere Informationen finden Sie unter „Mit dem Modelleditor arbeiten“ auf Seite 63.

  Anmerkung:

  Wenn Sie nicht über ausreichende Berechtigungen verfügen (das heißt, wenn Sie nur die LESEN-Berechtigung für "Regeln" (englisch: CEP management) haben anstatt der ADMIN-Berechtigung), dann enthält das Aktionsmenü den Befehl **Anzeigen** anstelle des Befehls **Bearbeiten**.

Die Instanzen eines Modells bearbeiten

Wenn einer oder mehrere Blöcke in einem Modell Vorlagenparameter verwenden (siehe auch „Vorlagenparameter verwalten“ auf Seite 78), dann können Sie verschiedene Instanzen dieses Modell erstellen.

Jede Instanz kann dann andere Werte für die Vorlagenparameter verwenden und kann unabhängig von den anderen Instanzen aktiviert werden. Die Instanzen werden im Instanzeditor definiert und aktiviert.

Anmerkung:

Die unten beschriebenen Aktionen können nur dann durchgeführt werden, wenn bereits Vorlagenparameter für das Model definiert wurden, das heißt, wenn auf der Karte für das Modell die Anzahl der definierten Instanzen zu sehen ist.

Die Instanzen eines Modells bearbeiten

- Auf der Registerkarte **Modelle** des Modellmanagers: klicken Sie auf der Vorderseite der Karte auf die Gesamtzahl der Instanzen.

  Alternativ können Sie auch eine der folgenden Aktionen ausführen:

  - Klicken Sie auf das Aktionsmenü des Modells und klicken Sie dann auf **Instanzen**.
  - Oder klicken Sie auf **Instanzenübersicht anzeigen**, um die Karte zu drehen, und klicken Sie dann auf der Rückseite der Karte auf die Schaltfläche **Instanzen bearbeiten**.

    Anmerkung:

    Instanzen bearbeiten ist nur dann sichtbar (und Sie können daher nur dann auf die Rückseite der Karte gelangen), wenn Instanzen vorhanden sind (unabhängig vom Status).

  Dies ruft den Instanzeditor auf. Weitere Informationen finden Sie unter „Mit dem Instanzeditor arbeiten“ auf Seite 91.

Ein Modell aktivieren

Ein Modell (oder eine Instanz) kann einen von zwei Zuständen haben. Der aktuelle Zustand (Status) wird immer auf der Karte eines Modells angezeigt:

- **Aktiv.** Dieser Status zeigt an, dass das Modell zur Zeit eingesetzt wird.
- **Inaktiv.** Dieser Status zeigt an, dass das Modell zur Zeit nicht eingesetzt wird.

Die Eingaben, die ein Modell empfängt und was mit den Ausgaben passiert, hängt von dem Modus ab, in dem sich das Modell gerade befindet. Jedes Modell kann auf einen der folgenden Modi gesetzt werden:

- **Entwurf.** Das Modell befindet sich noch in der Entwicklung. (Neue Modelle werden im Modus "Entwurf" erstellt.)

- **Test.** Dieser Modus ist nur bei Modellen erlaubt, die ein einziges Gerät benutzen. Wenn der Status des Modells "Aktiv" ist, dann wird das Modell zur Zeit auf dem Apama-Korrelator eingesetzt und die von dem Gerät kommenden Messwerte und Ereignisse werden verarbeitet. Die Ausgabe des Modells wird aber nur gespeichert (und als `Operation`- oder `Measurement`-Objekt eines „virtuellen Geräts“ aufgezeichnet) und *nicht* an das Gerät zurückgesendet.

  Anmerkung:

  Der Testmodus wird nicht unterstützt bei einem Modell, das einen benutzerdefinierten Block enthält, der Eingabedaten verarbeitet und außerdem auch Ausgabedaten erzeugt. Benutzerdefinierte Blöcke werden mit dem Block SDK erstellt; siehe auch „Eigene Blöcke erstellen“ auf Seite 45.

- **Simulation.** Dieser Modus ist nur bei Modellen erlaubt, die ein einziges Gerät benutzen. Wenn der Status des Modells "Aktiv" ist, dann wird das Modell zur Zeit auf dem Apama-Korrelator eingesetzt. Es verwendet historische Eingabedaten; diese werden in Echtzeit von den zuvor den empfangenen Daten durchgespielt. Die Ausgabe des Modells wird aber nur gespeichert (und als `Operation`- oder `Measurement`-Objekt eines „virtuellen Geräts“ aufgezeichnet) und *nicht* an das Gerät zurückgesendet. Um eine Simulation zu starten, müssen Sie den Zeitraum definieren, aus dem die Eingabedaten verwendet werden sollen. Wenn alle Daten aus diesem Zeitraum wiedergegeben wurden, dann wird die Datenverarbeitung für das Modell automatisch beendet und der Modellstatus wird auf "Inaktiv" zurückgesetzt. Die Zeitstempel der historischen Dateneinträge bleiben zum leichteren Vergleich von Simulationsläufen unverändert. Siehe auch „Modellsimulation“ auf Seite 143.

- **Produktion.** Wenn der Status des Modells "Aktiv" ist, dann wird das Modell zur Zeit auf dem Apama-Korrelator eingesetzt und die von den Geräten kommenden Messwerte und Ereignisse werden verarbeitet. Die Ausgabe des Modells wird gespeichert und an die Geräte zurückgesendet.

Ein Modell im Entwurfsmodus kann sich immer nur im Status "Inaktiv" befinden. Ein Modell im Test-, Simulations- oder Produktionsmodus kann sich entweder im Status "Aktiv" oder "Inaktiv" befinden.

Anmerkung:

Die oben genannten Informationen zum Status und Modus eines Modell gelten in ähnlicher Weise für die Instanzen eines Vorlagenmodells. Die unten beschriebene Anweisung gilt jedoch nur für Modelle ohne Vorlagenparameter. Wenn Sie die Instanzen eines Vorlagenmodells aktivieren möchten, dann finden Sie die entsprechende Anweisung unter „Eine Instanz aktivieren“ auf Seite 94.

Wenn ein Modell aus einer JSON-Datei importiert wird, wird es immer als inaktives Modell importiert.

Ein Modell aktivieren

1. Auf der Registerkarte **Modelle** des Modellmanagers: klicken Sie auf der Karte für das Modell, das Sie aktivieren möchten, auf das Dropdown-Menü und wählen Sie eine der folgenden Optionen aus: **Produktion**, **Test** oder **Simulation**.

2. Wenn Sie den Simulationsmodus ausgewählt haben, klicken Sie auf das nun angezeigte Kalendersymbol, geben Sie den gewünschten Zeitraum an und klicken Sie auf **Anwenden**. Siehe auch „Simulationsparameter“ auf Seite 145.

3. Wenn die Umschaltfläche gerade **Inaktiv** anzeigt, klicken Sie auf diese Umschaltfläche, um den Status in **Aktiv** zu ändern. Für den Simulationsmodus können Sie den Status nur dann auf **Aktiv** setzen, wenn Sie einen gültigen Zeitraum angegeben haben.

Ein Modell deaktivieren

Sie können jedes Modell deaktivieren, das sich gerade im Produktions-, Test- oder Simulationsmodus befindet und dessen Umschaltfläche **Aktiv** anzeigt.

Wenn Sie ein Modell deaktivieren, dann wird das Modell gestoppt und eingehende Daten werden nicht mehr verarbeitet. Jeder im Modell aufgebaute Zustand geht verloren. Für den Simulationsmodus bedeutet dies, dass das Modell gestoppt wird, bevor alle historischen Daten aus dem angegebenen Zeitraum durchgespielt wurden.

Anmerkung:

Wenn Sie die Instanzen eines Vorlagenmodells deaktivieren möchten, dann finden Sie die entsprechende Anweisung unter „Eine Instanz deaktivieren“ auf Seite 95.

Ein Modell deaktivieren

- Auf der Registerkarte **Modelle** des Modellmanagers: klicken Sie auf der Karte für das Modell, das Sie deaktivieren möchten, auf die Umschaltfläche, so dass **Inaktiv** auf dieser Umschaltfläche angezeigt wird.

Ein Modell duplizieren

Sie können jedes Modell duplizieren, das zur Zeit im Modellmanager aufgelistet wird.

Das Duplikat erhält denselben Namen wie das Originalmodell, gefolgt von dem Nummernzeichen (#) und einer Zahl. Wenn der Name des Originalmodells zum Beispiel „Mein Modell“ ist, dann ist der Name des ersten Duplikats „Mein Modell #1“. Die Zahl im Modellnamen wird mit jedem nachfolgenden Duplikat um eins erhöht. Das Duplikat erhält die gleiche Beschreibung wie das Originalmodell. Es wird empfohlen, dass Sie das Duplikat bearbeiten und dem Modell einen sinnvollen Namen und eine aussagekräftige Beschreibung geben.

Ein Modell duplizieren

- Auf der Registerkarte **Modelle** des Modellmanagers: klicken Sie auf das Aktionsmenü des Modells, das Sie duplizieren möchten, und klicken Sie dann auf **Duplizieren**.

  Im Modellmanager wird sofort eine Karte für das Duplikat angezeigt.

Ein Modell exportieren

Sie können jedes Modell exportieren, das zur Zeit im Modellmanager aufgelistet wird. Dies ist zum Beispiel dann hilfreich, wenn Sie ein Modell vom aktuellen Cumulocity IoT-Mandanten auf einen anderen Mandanten übertragen möchten. Das Modell wird im JSON-Format gespeichert.

Ein Modell exportieren

- Auf der Registerkarte **Modelle** des Modellmanagers: klicken Sie auf das Aktionsmenü des Modells, das Sie exportieren möchten, und klicken Sie dann auf **Exportieren**.

  Das anschließende Verhalten ist von Ihrem Browser abhängig. Das Modell wird normalerweise in das Download-Verzeichnis Ihres Browsers exportiert.

Ein Modell importieren

Sie können ein Modell importieren, das zuvor (im JSON-Format) exportiert wurde. Dies ist zum Beispiel dann hilfreich, wenn Sie ein Modell von einem anderen Cumulocity IoT-Mandanten importieren möchten.

Ein Modell importieren

1. Auf der Registerkarte **Modelle** des Modellmanagers: klicken Sie in der Symbolleiste auf **Modell importieren**.
2. Navigieren Sie im anschließend angezeigten Dialogfeld zu dem Ort, an dem das zu importierende Modell gespeichert ist.
3. Wählen Sie das Modell aus und klicken Sie auf **Öffnen**.

   Im Modellmanager wird nun eine Karte für das importierte Modell angezeigt.

Ein Modell entfernen

Sie können jedes Modell entfernen, das zur Zeit im Modellmanager aufgelistet wird. Wenn das Modell gerade aktiv ist, dann wird es zuerst deaktiviert und danach entfernt.

Ein Modell entfernen

1. Auf der Registerkarte **Modelle** des Modellmanagers: klicken Sie auf das Aktionsmenü des Modells, das Sie entfernen möchten, und klicken Sie dann auf **Entfernen**.

2. Klicken Sie im anschließend angezeigten Dialogfeld auf **Entfernen**, um das Entfernen zu bestätigen.

Alle Modelle neu laden

Sie können die Anzeige aktualisieren, um alle Änderungen anzuzeigen, die andere Benutzer seit dem Laden der Seite vorgenommen haben, oder um festzustellen, ob die aktiven Modelle in einen fehlerhaften Zustand übergegangen sind.

Alle Modelle neu laden

- Auf der Registerkarte **Modelle** des Modellmanagers: klicken Sie in der Symbolleiste auf **Neu laden**.

Ein Beispiel anzeigen

Die Beispiele werden immer im schreibgeschützen Modus angezeigt. Sie können den Inhalt jedes Beispiels anzeigen, das zur Zeit im Modellmanager aufgelistet wird.

Sie können zum Beispiel die Blockparameter überprüfen oder die Dokumentation für jeden Block anzeigen, der in dem Beispiel verwendet wird. Sie können in dem Beispiel genauso navigieren und zoomen wie in einem normalen Modell, aber Sie können nichts hinzufügen oder verändern. Sie können jedoch ein neues Modell auf der Grundlage des Beispiels erstellen (siehe auch „Ein Modell aus einem Beispiel erstellen“ auf Seite 60).

Ein Beispiel anzeigen

- Auf der Registerkarte **Beispiele** des Modellmanagers: klicken Sie einfach auf die Karte, die für das Beispiel angezeigt wird.

  Oder klicken Sie auf das Aktionsmenü des Beispiels und klicken Sie dann auf **Anzeigen**.

Ein Modell aus einem Beispiel erstellen

Sie können aus jedem Beispiel, das zur Zeit im Modellmanager aufgelistet wird, ein neues Modell erstellen. Das neue Modell erhält denselben Namen, dieselbe Beschreibung und dieselben Tags wie das Beispiel.

Anmerkung:
Sie müssen das neue Modell speichern, damit es im Modellmanager aufgelistet wird. Wenn es bereits ein Modell mit diesem Namen gibt, dann werden Sie aufgefordert, das neue Modell unter einem anderen Namen zu speichern.

Ein Modell aus einem Beispiel erstellen

- Auf der Registerkarte **Beispiele** des Modellmanagers: klicken Sie auf das Aktionsmenü des Beispiels, auf dessen Grundlage Sie ein neues Modell erstellen wollen, und klicken Sie dann auf **Modell aus Beispiel erstellen**.

  Oder wenn das Beispiel zur Zeit im Modeleditor angezeigt wird: klicken Sie in der Symbolleiste auf **Modell aus Beispiel erstellen**.

  Das neue Modell wird sofort im Modelleditor angezeigt und Sie können nun jeden Aspekt des Modells ändern.

5 Mit dem Modelleditor arbeiten

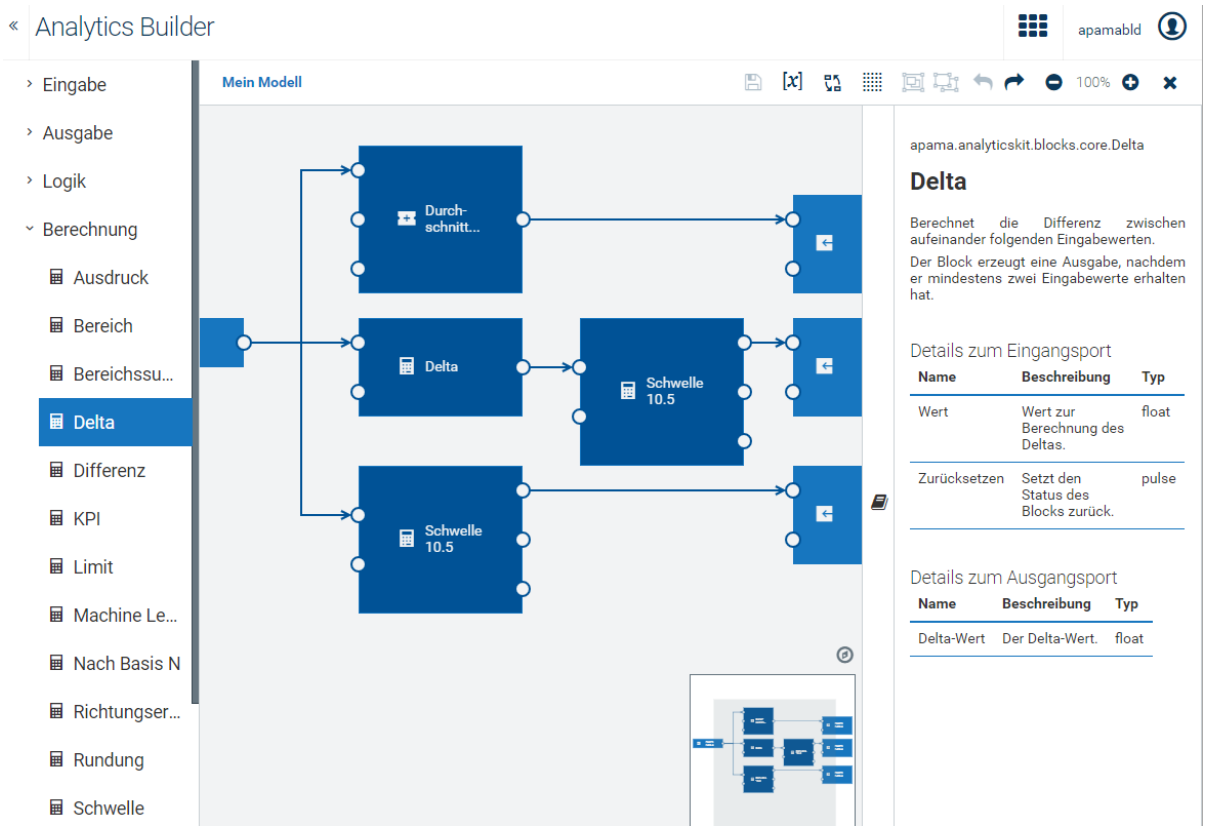
- Die Benutzeroberfläche des Modelleditors 64
- Mit Modellen arbeiten 65
- Mit Blöcken und Verbindungen arbeiten 67
- Mit Gruppen arbeiten 81
- Die Arbeitsfläche verwalten 87

Die Benutzeroberfläche des Modelleditors

Mit dem Modelleditor können Sie Analysemodelle grafisch erstellen. Er wird aufgerufen, wenn Sie ein Modell im Modellmanager erstellen oder bearbeiten. Siehe auch „Ein neues Modell erstellen“ auf Seite 54 und „Ein bestehendes Modell bearbeiten“ auf Seite 55.

Anmerkung:

Der Modelleditor wird auch aufgerufen, wenn Sie ein Beispiel anzeigen. In diesem Fall wird der Modelleditor jedoch im schreibgeschützen Modus aufgerufen und die Palette auf der linken Seite wird nicht angezeigt. Siehe auch „Ein Beispiel anzeigen“ auf Seite 60.



Die Palette auf der linken Seite enthält die Blöcke, die Sie Ihrem Modell hinzufügen können. Sie hat mehrere auf- und zuklappbare Kategorien für die verschiedenen Arten von Blöcken.

Die Arbeitsfläche in der Mitte ist der Bereich, in dem Sie Ihr Modell „zeichnen“. Sie ziehen die Blöcke aus der Palette auf die Arbeitsfläche, definieren die Parameter für die Blöcke und verbinden die Blöcke miteinander. Der Inhalt der Arbeitsfläche wird an einem Raster ausgerichtet (siehe auch „Das Raster anzeigen und ausblenden“ auf Seite 89).

Die Modellübersicht rechts unten auf der Arbeitsfläche zeigt das gesamte Modell. Dies ist hilfreich, wenn Ihr Modell zu groß ist und nicht in den zur Zeit sichtbaren Bereich der Arbeitsfläche passt. Siehe auch „In einem großen Modell navigieren“ auf Seite 87.

In der Dokumentationsanzeige auf der rechten Seite können Sie Referenzinformationen für den zur Zeit ausgewählten Block sehen. Siehe auch „Die Dokumentation für einen Block anzeigen“ auf Seite 72.

ACHTUNG:

Ihre Änderungen werden nur dann gespeichert, wenn Sie auf klicken (siehe auch „Ein Modell speichern“ auf Seite 66). Der Editor warnt Sie, wenn Sie versuchen, vom Editor weg zu navigieren und nicht gespeicherte Änderungen vorhanden sind. Sie sollten jedoch immer sicherstellen, dass Ihre Änderungen gespeichert wurden, bevor Sie den Browser vom Netzwerk trennen oder einen Laptop in den Ruhezustand versetzen.

Mit Modellen arbeiten

Übersicht der Schritte für das Erstellen eines Modells

Dieser Abschnitt gibt Ihnen eine kurze Übersicht darüber, wie Sie ein neues Modell erstellen und gestalten. Ausführlichere Informationen finden Sie in den Abschnitten, auf die in den folgenden Schritten verwiesen wird.

Sie erstellen und gestalten ein Modell wie folgt:

1. Klicken Sie im Modellmanager auf **Neues Modell**. Geben Sie im anschließend angezeigten Dialogfeld einen Modellnamen ein. Siehe auch „Ein neues Modell erstellen“ auf Seite 54.
2. Ziehen Sie im Modelleditor die erforderlichen Blöcke aus der Palette auf die Arbeitsfläche. Siehe auch „Einen Block hinzufügen“ auf Seite 67.
3. Lesen Sie bei Bedarf die Blockdokumentation. Siehe auch „Die Dokumentation für einen Block anzeigen“ auf Seite 72.
4. Definieren Sie die Parameter des Blocks im Blockparametereditor. Siehe auch „Die Parameter eines Blocks editieren“ auf Seite 69.
5. Verknüpfen Sie die Blöcke mit Verbindungen. Siehe auch „Zwei Blöcke verbinden“ auf Seite 74.
6. Speichern Sie Ihre Änderungen. Siehe auch „Ein Modell speichern“ auf Seite 66.

   Anmerkung:
   Im Modellmanager werden nur die Modelle angezeigt, die explizit gespeichert wurden. Wenn Sie ein neues Modell erstellen und den Modelleditor verlassen, ohne das Modell zu speichern, dann wird es nicht im Modellmanager angezeigt und all Ihre Änderungen sind verloren.
7. Verlassen Sie den Modelleditor. Das bringt Sie automatisch zum Modellmanager zurück. Siehe auch „Den Modelleditor verlassen“ auf Seite 67.
8. Ein neu erstelltes Modell wird im Modellmanager automatisch in den Entwurfsmodus gesetzt. Wenn Sie das Modell testen, simulieren oder in der Produktion einsetzen möchten, dann finden Sie weitere Informationen in „Ein Modell aktivieren“ auf Seite 56.

Detaillierte Hintergrundinformationen, einschließlich Einschränkungen, finden Sie unter „Verbindungen und Blöcke“ auf Seite 99.

Name, Beschreibung und Tags eines Modells ändern

Sie können jedes Modell umbenennen, das Sie gerade im Modelleditor bearbeiten, und Sie können dann auch die Beschreibung des Modells ändern.

Außerdem können Sie Tags hinzufügen oder entfernen. Tags sind hilfreich, wenn Sie die Modelle im Modellmanager filtern und nur die Modelle anzeigen möchten, für die ein bestimmtes Tag definiert wurde (siehe auch „Modelle und Beispiele filtern“ auf Seite 52).

Name, Beschreibung und Tags eines Modells ändern

1. Klicken Sie im Modelleditor auf den Modellnamen. Der Modellname wird links in der Symbolleiste angezeigt.

2. Geben Sie im anschließend angezeigten Dialogfeld "Modellkonfiguration" einen neuen eindeutigen Namen für das Modell an, ändern Sie die Beschreibung und/oder ändern Sie die Tags.

   Um ein Tag hinzuzufügen, geben Sie einfach seinen Namen ein und drücken die Eingabe- oder Tabulatortaste. Das Tag wird dann in einem farbigen Rechteck angezeigt. Um ein Tag zu entfernen, klicken Sie auf das im Rechteck angezeigte . Das Dialogfeld verhindert, dass Sie für ein Modell doppelte Tags eingeben; wenn Sie einen solchen Tag-Namen eingeben, wird das doppelte Tag nicht hinzugefügt und das bestehende Tag blinkt einmal.

3. Klicken Sie auf **OK**.

Ein Modell speichern

Wenn Sie ein Modell im Modelleditor speichern, wird es im Cumulocity IoT-Inventar für Ihren Mandanten im JSON-Format abgelegt.

Wichtig:
Es kann vorkommen, dass Sie und ein anderer Benutzer dasselbe Modell gleichzeitig bearbeiten. In diesem Fall werden die zuletzt gespeicherten Änderungen im Inventar abgelegt. Ihre Änderungen werden also möglicherweise durch ein späteres Speichern durch einen anderen Benutzer überschrieben.

Ein Modell speichern

- Klicken Sie in der Symbolleiste des Modelleditors auf .

  Diese Schaltfläche in der Symbolleiste ist nur dann aktiviert, wenn Änderungen am Modell vorgenommen wurden und wenn das Modell einen Namen erhalten hat.

Den Modelleditor verlassen

Wenn Sie den Modelleditor mit der entsprechenden Schaltfläche in der Symbolleiste verlassen, kehren Sie zum Modellmanager zurück. Sie können dann zum Beispiel ein anderes Modell bearbeiten oder den Modus oder Status des aktuellen Modells ändern.

ACHTUNG:
Alle nicht gespeicherten Änderungen gehen verloren, wenn Sie zu einer anderen URL navigieren oder das Browserfenster schließen.

Den Modelleditor verlassen

- Klicken Sie in der Symbolleiste des Modelleditors auf ✖.

  Falls noch nicht gespeicherte Änderungen vorhanden sind, werden Sie gefragt, ob Sie diese speichern oder verwerfen möchten.

Mit Blöcken und Verbindungen arbeiten

Einen Block hinzufügen

Die Blöcke in der Palette sind in verschiedene Kategorien unterteilt. Die Blöcke in den Kategorien **Eingabe** und **Ausgabe** stehen für Geräte, Gerätegruppen und/oder (wenn das Suchfeld verwendet wird) beliebige Assets, die im Cumulocity IoT-Inventar registriert wurden (diese werden in der Device Management-Anwendung visualisiert). Andere Kategorien enthalten Blöcke, mit denen man dem Modell zum Beispiel logische Operationen oder Berechnungen hinzufügen kann.

Wenn Sie den Mauszeiger über einen Block in der Palette bewegen, wird ein Tooltip angezeigt, der den Zweck des Blocks kurz erläutert. Der Tooltip zeigt auch den kompletten Namen des Blocks an.

In den Kategorien **Eingabe** und **Ausgabe** stehen unterhalb der jeweiligen Baumstrukturen horizontale Bildlaufleisten zur Verfügung. Diese Bildlaufleisten - und auch die Tooltips - sind hilfreich, wenn der Name eines Geräts, einer Gerätegruppe oder eines Assets nicht vollständig in der Palette angezeigt wird.

Detaillierte Informationen zu jedem Block finden Sie in der Blockreferenz, die in der Dokumentationsanzeige zu sehen ist; siehe auch „Die Dokumentation für einen Block anzeigen“ auf Seite 72.

Einen Block hinzufügen

1. Klappen Sie in der Palette des Modelleditors die Kategorie auf, die den Block enthält, den Sie hinzufügen möchten.

2. Wenn Sie die Kategorie **Eingabe** oder **Ausgabe** aufklappen, werden zunächst die Geräte und Gerätegruppen angezeigt, die im Cumulocity IoT-Inventar registriert sind.

Standardmäßig werden 10 Geräte und 10 Gerätegruppen angezeigt; diese sind alphabetisch sortiert. Bei einem großen Inventar müssen Sie auf **Mehr laden** klicken, um weitere Geräte und Gerätegruppen anzuzeigen.

Die Palette spiegelt die Eltern/Kind-Hierarchie im Cumulocity IoT-Inventar wider. Die Liste der Geräte enthält alle definierten Kind-Geräte, und die Liste der Gerätegruppen enthält alle definierten Untergruppen. Diese stehen in auf- und zuklappbaren Knoten zur Verfügung.

Die Kategorien **Eingabe** und **Ausgabe** haben beide ein Suchfeld. Mit diesen Suchfeldern können Sie alle Assets im Cumulocity IoT-Inventar anzeigen, die Ihren Suchkriterien entsprechen. Dies umfasst Geräte, Gerätegruppen und auch Managed-Objects. Bei der Suche wird zwischen Groß- und Kleinschreibung unterschieden. Sie können einen Namen oder einen Teil eines Namens eingeben. Die von Ihnen eingegebenen Zeichen können an einer beliebigen Position innerhalb des Namens enthalten sein. Geben Sie beispielsweise das Wort "test" ein, um nur die Assets anzuzeigen, die dieses Wort im Namen haben. Die Eingabetaste brauchen Sie nicht drücken; die Liste wird mit jedem von Ihnen eingegebenen Zeichen aktualisiert. Das Suchergebnis ist alphabetisch sortiert. Bei einem großen Suchergebnis müssen Sie auf **Mehr laden** klicken, um weitere Assets anzuzeigen.

Die maximale Anzahl der angezeigten Geräte, Gerätegruppen und/oder Assets wird durch die Mandantenoptionen bestimmt. Weitere Informationen finden Sie in „Die Anzahl der angezeigten Geräte, Gerätegruppen und/oder Assets konfigurieren“ auf Seite 141.

Die Assets, die bei der Suche angezeigt werden, werden auch durch die Mandantenoptionen bestimmt. Sie können die Suche so einschränken, dass nur Assets eines bestimmten Typs angezeigt werden. Weitere Informationen finden Sie unter „Nach Eingabe- und Ausgabe-Assets suchen“ auf Seite 141.

Wenn Sie eine andere Kategorie aufklappen, dann werden Ihre Suchkriterien im Datenspeicher behalten. Dies ist hilfreich, wenn Sie kürzere Listen für die Kategorien **Eingabe** und **Ausgabe** erzeugen möchten, und jede Kategorie hierbei individuelle Einträge haben soll, die Ihren Suchkriterien entsprechen.

Die Kategorie **Ausgabe** bietet einen speziellen Block mit dem Namen **Trigger-Gerät** zur Auswahl an. Dieser Block sendet die Ausgabe zurück an das Gerät, das die Ausgabe ausgelöst hat.

3. Ziehen Sie den Block aus der Palette und legen Sie ihn auf der Arbeitsfläche ab.

   Wenn Sie den Block auf einem vorhandenen Block auf der Arbeitsfläche ablegen, dann wird der neue Block über diesem Block erstellt. Wenn Sie den Block auf einer zugeklappten Gruppe ablegen, dann wird der neue Block unterhalb dieser Gruppe erstellt. In beiden Fällen sollten Sie den neuen Block auf einen freien Bereich der Arbeitsfläche verschieben. Siehe auch „Einen Block verschieben“ auf Seite 73.

   Wenn Sie den Block auf einer aufgeklappten Gruppe ablegen (wobei der Inhalt der Gruppe sichtbar ist), dann wird der neue Block dieser Gruppe hinzugefügt. Weitere Informationen zu Gruppen finden Sie unter „Mit Gruppen arbeiten“ auf Seite 81.

   Anmerkung:

In der Kategorie **Ausgabe** werden zwar Gerätegruppen angezeigt, es ist aber nicht möglich diese auf die Arbeitsfläche zu ziehen, weil eine Gruppenausgabe nicht erlaubt ist.

4. Definieren Sie alle erforderlichen Parameter für den Block. Siehe „Die Parameter eines Blocks editieren“ auf Seite 69.

Anmerkung:

Der Blockparametereditor wird automatisch angezeigt, wenn Sie einen Block hinzufügen, für den Parameter angegeben werden müssen. Es wird jedoch nicht angezeigt, wenn für den Block keine Parameter erforderlich sind (wie zum Beispiel beim **ODER**-Block).

Die Parameter eines Blocks editieren

Die meisten Blöcke (aber nicht alle) haben Parameter, die Sie entsprechend Ihren Anforderungen einstellen müssen.

Wenn auf einem Ein- oder Ausgabeblock auf der Arbeitsfläche „Fehlt“ angezeigt wird, dann bedeutet dies, dass das definierte Gerät, die definierte Gerätegruppe oder das definierte Asset nicht im Cumulocity IoT-Inventar gefunden werden kann. Sie sollten dann entweder zum Cumulocity IoT-Inventar gehen und sicherstellen, dass dieses Gerät registriert ist oder dass die Gerätegruppe oder das Asset vorhanden ist, oder Sie sollten ein anderes vorhandenes Gerät, eine andere vorhandene Gerätegruppe oder ein anderes vorhandenes Asset im Blockparametereditor auswählen (siehe unten).

Die Beschriftungen von einigen Blöcken auf der Arbeitsfläche zeigen den Wert des wichtigsten Parameters. Der **Ausdruck**-Block zeigt zum Beispiel den definierten Ausdruck und der **Zeitverzögerung**-Block zeigt die definierte Verzögerung in Sekunden.

Der Blockparametereditor enthält auch Befehle zum Duplizieren und Entfernen des ausgewählten Blocks. Weitere Informationen finden Sie unter „Einen Block duplizieren“ auf Seite 73 und „Einen Block oder eine Verbindung entfernen“ auf Seite 75.

Bei den Ein- und Ausgabeblöcke können Sie die verwendeten Geräte, Gerätegruppen oder Assets global ersetzen. Ausführliche Informationen finden Sie unter „Geräte, Gerätegruppen und Assets ersetzen“ auf Seite 76.

Die Parameter eines Blocks editieren

1. Klicken Sie auf der Arbeitsfläche des Modelleditors mit der *linken* Maustaste auf den Block, den Sie bearbeiten möchten.

   Der Blockparametereditor wird angezeigt. Er enthält Eingabefelder für alle Parameter, die für diesen Block angegeben werden können.

2. Bei den Eingabe- und Ausgabeblöcken zeigt der Blockparametereditor das Dropdown-Listenfeld **Blocktyp** an. Wählen Sie den Blocktyp aus, der Ihren Anforderungen entspricht.

Anmerkung:

Folgendes gilt, wenn Sie den Blocktyp bei einem Block ändern, der bereits mit einem oder mehreren anderen Blöcken verbunden ist: Wenn der neue Blocktyp andere Portnamen hat (wenn sich der Portname zum Beispiel von **Wert** in **Wert 1** ändert), dann werden die vorhandenen Verbindungen zu/von den geänderten Ports entfernt. Dies geschieht, weil ein geänderter Portname die vorhandene Verbindung ungültig machen würde.

3. Bei den Eingabe- und Ausgabeblöcken können Sie ein anderes Gerät (für die Eingabe oder Ausgabe), eine andere Gerätegruppe (nur für die Eingabe) oder ein anderes Asset (wenn das Suchfeld verwendet wird) aus einem Dialogfeld auswählen.

   Standardmäßig werden 10 Geräte und 10 Gerätegruppen angezeigt; diese sind alphabetisch sortiert. Bei einem großen Inventar müssen Sie auf **Mehr laden** klicken, um weitere Geräte und Gerätegruppen anzuzeigen.

   Die Baumstruktur in dem Dialogfeld spiegelt die Eltern/Kind-Hierarchie im Cumulocity IoT-Inventar wider. Die Liste der Geräte enthält alle definierten Kind-Geräte, und die Liste der Gerätegruppen enthält alle definierten Untergruppen. Diese stehen in auf- und zuklappbaren Knoten zur Verfügung. Bei den Ausgabeblöcken können Sie auch das **Trigger-Gerät** auswählen.

   Mit dem Suchfeld können Sie alle beliebigen Assets im Cumulocity IoT-Inventar anzeigen, die Ihren Suchkriterien entsprechen. Dies umfasst Geräte, Gerätegruppen und auch Managed-Objects. Bei der Suche wird zwischen Groß- und Kleinschreibung unterschieden. Die von Ihnen eingegebenen Zeichen können an einer beliebigen Position innerhalb des Namens enthalten sein. Die Baumstruktur wird mit jedem von Ihnen eingegebenen Zeichen aktualisiert. Bei einem großen Suchergebnis müssen Sie auf **Mehr laden** klicken, um weitere Assets anzuzeigen.

   Klicken Sie die Schaltfläche **Verwenden** (diese wird angezeigt, wenn Sie mit der Maus über einen Eintrag fahren), um das gewünschte Gerät, die Gerätegruppe oder das Asset auszuwählen. Für Ausgabeblöcke können Sie keine Gerätegruppe auswählen, Sie können aber jedes beliebige Gerät innerhalb dieser Gruppe auswählen.

   Die maximale Anzahl der angezeigten Geräte, Gerätegruppen und/oder Assets wird durch die Mandantenoptionen bestimmt. Weitere Informationen finden Sie in „Die Anzahl der angezeigten Geräte, Gerätegruppen und/oder Assets konfigurieren“ auf Seite 141.

   Die Assets, die bei der Suche angezeigt werden, werden auch durch die Mandantenoptionen bestimmt. Sie können die Suche so einschränken, dass nur Assets eines bestimmten Typs angezeigt werden. Weitere Informationen finden Sie unter „Nach Eingabe- und Ausgabe-Assets suchen“ auf Seite 141.

4. Anstatt einen Wert für einen Blockparameter einzugeben, können Sie auch einen Vorlagenparameter verwenden. Dadurch können in den verschiedenen Instanzen des Modells unterschiedliche Werte für diesen Blockparameter verwendet werden (weitere Informationen finden Sie unter „Mit dem Instanzeditor arbeiten“ auf Seite 91). Erstellen Sie einen Vorlagenparameter eines passenden Typs im Dialogfeld "Vorlagenparameter" (siehe „Vorlagenparameter verwalten“ auf Seite 78), schalten Sie den Blockparameter dann auf die Verwendung eines Vorlagenparameters um (siehe unten) und wählen Sie den gewünschten

Vorlagenparameter aus dem Dropdown-Listenfeld aus. Oder erstellen Sie den Vorlagenparameter direkt im Blockparametereditor (siehe unten).

Der Blockparametereditor bietet die folgenden Optionen in einem Dropdown-Listenfeld an:

- Wenn diese Option ausgewählt ist, dann können Sie mit dem nebenstehenden Steuerelement einen Wert für diesen Parameter angeben. Dieser Wert wird im Blockparametereditor validiert.

- Wenn diese Option ausgewählt ist, dann können Sie aus dem nebenstehenden Dropdown-Listenfeld einen Vorlagenparameter auswählen. Sie können nur einen Vorlagenparameter auswählen, der vom selben Typ ist wie der aktuelle Blockparameter. Vorlagenparameter ungeeigneter Typen stehen nicht zur Auswahl. Vorlagenparameter werden im Blockparametereditor nicht validiert.

  Wenn Sie einen neuen Vorlagenparameter direkt im Blockparametereditor hinzufügen möchten, geben Sie einen Namen in das Textfeld des Dropdown-Listenfelds ein. Sobald Sie mit der Eingabe beginnen und noch kein Vorlagenparameter mit diesem Namen vorhanden ist, wird unter dem Textfeld die Option **Vorlagenparameter hinzufügen "*Name*"** angezeigt. Klicken Sie auf diese Option, um den neuen Vorlagenparameter hinzuzufügen und ihn somit im Dialogfeld "Vorlagenparameter" verfügbar zu machen. Der neue Vorlagenparameter hat denselben Typ, sowie dieselben optionalen und Standardwerte wie der Blockparameter. Wenn bereits ein Vorlagenparameter mit dem von Ihnen angegebenen Namen existiert, jedoch mit einem inkompatiblen Typ, dann werden der Name und der Typ unter dem Textfeld angezeigt; der Vorlagenparameter kann jedoch nicht ausgewählt werden.

5. Bei einigen Blöcken (wie zum Beispiel beim **Bereichssuche**-Block) zeigt der Blockparametereditor Textfelder für die Eingabe von Schlüssel-Wertepaaren an. Wenn Sie weitere Schlüssel-Wertepaare eingeben möchten, klicken Sie auf **Zeile hinzufügen**. Das Schlüssel-Wertepaar in der ersten Zeile wird zuerst verarbeitet. Sie können eine Zeile mit dem neben dieser Zeile angezeigten Steuerelement ⇕ an eine andere Position ziehen. Nicht mehr benötigte Zeilen können Sie entfernen, indem Sie auf das ✖ neben dieser Zeile klicken. Leere Zeilen werden automatisch entfernt, wenn Sie den Blockparametereditor verlassen.

6. Geben Sie alle erforderlichen Parameter an.

   Ausführliche Referenzinformationen für jeden Block (und Blocktyp) finden Sie in der Dokumentationsanzeige. Siehe auch „Die Dokumentation für einen Block anzeigen“ auf Seite 72.

   Ihre Eingabe wird im Datenspeicher behalten, wenn Sie den Blockparametereditor verlassen (zum Beispiel, wenn Sie auf einen anderen Block oder auf die Arbeitsfläche klicken).

   Anmerkung:

   Denken Sie daran, dass Ihre Änderungen nur dann dauerhaft ins Inventar geschrieben werden, wenn Sie das Modell speichern. Siehe auch „Ein Modell speichern“ auf Seite 66.

Die Dokumentation für einen Block anzeigen

In der Dokumentationsanzeige können Sie detaillierte Informationen zum aktuell ausgewählten Block sehen. Hier wird die sogenannte *Blockreferenz* angezeigt, in der die Parameter und die Ein- und Ausgangsports eines Blocks beschrieben sind. Sie können die Größe der Dokumentationsanzeige ändern und die Anzeige ein- und ausblenden.

Anmerkung:

Sie können die Blockreferenz auch direkt in dieser Dokumentation einsehen. Siehe Blockreferenz.

Die Dokumentation für einen Block anzeigen

1. Klicken Sie im Modelleditor auf den Block, für den Sie die Dokumentation anzeigen möchten. Sie können dies in der Palette oder auf der Arbeitsfläche tun.

2. Wenn das Dropdown-Listenfeld **Blocktyp** im Blockparametereditor angezeigt wird, wählen Sie den Blocktyp, für den die Dokumentation angezeigt werden soll.

3. Wenn die Dokumentationsanzeige zur Zeit nicht sichtbar ist, klicken Sie auf den Bereich mit dem Symbol (rechts neben der Arbeitsfläche) um die Dokumentationsanzeige einzublenden. Durch erneutes Klicken auf diesen Bereich wird die Dokumentationsanzeige wieder ausgeblendet.

4. Wenn Sie die Größe der Dokumentationsanzeige ändern (zum Beispiel vergrößern) möchten, dann bewegen Sie den Mauszeiger über den Bereich mit dem oben genannten Symbol. Klicken und halten Sie die Maustaste gedrückt, und ziehen Sie die Maus dann nach links oder rechts (um die Dokumentationsanzeige zu vergrößern oder zu verkleinern).

Blöcke und Verbindungen auswählen

Wenn Sie einen oder mehrere Blöcke auf der Arbeitsfläche des Modelleditors verschieben, duplizieren oder entfernen möchten, müssen Sie die erforderlichen Blöcke zuerst auswählen.

Um einen einzelnen Block auf der Arbeitsfläche auszuwählen, klicken Sie einfach auf den Block. Bei einem Block hängt das anschließende Verhalten von der verwendeten Maustaste ab:

- Wenn Sie den Block mit der *linken* Maustaste anklicken, dann wird der Block ausgewählt und der Blockparametereditor wird angezeigt (siehe auch „Die Parameter eines Blocks editieren“ auf Seite 69).
- Wenn Sie den Block mit der *rechten* Maustaste anklicken, dann wird der Block nur ausgewählt. Der Blockparametereditor wird nicht angezeigt. Dies ist zum Beispiel dann hilfreich, wenn der Editor beim Hinzufügen einer Verbindung zu einem anderen Block im Weg wäre.

Um eine einzelne Verbindungen auszuwählen, müssen Sie nur auf die Verbindung klicken (in diesem Fall können Sie eine beliebige Maustaste verwenden).

Um mehrere Blöcke und/oder Verbindungen gleichzeitig auszuwählen, führen Sie einen der folgenden Schritte aus:

- Drücken Sie die Strg-Taste und klicken Sie auf jeden Block, den Sie auswählen möchten und/oder auf jede Verbindung, die Sie auswählen möchten.
- Oder wenn Sie einen Bereich mit mehreren Blöcken und Verbindungen auswählen möchten, klicken Sie mit der linken Maustaste auf einen leeren Bereich der Arbeitsfläche und halten Sie die Maustaste gedrückt. Warten Sie, bis der Mauszeiger als Kreuz angezeigt wird. Ziehen Sie dann die Maus, um den gewünschten Bereich auszuwählen. Lassen Sie die Maustaste los, wenn alle erforderlichen Blöcke und Verbindungen ausgewählt wurden.
- Oder um alle Blöcke und Verbindungen auszuwählen, drücken Sie Strg+A.

Um die Auswahl aufzuheben:

- Drücken Sie die Strg-Taste und klicken Sie auf den aktuell ausgewählten Block oder die Verbindung.
- Oder klicken Sie auf eine leere Stelle der Arbeitsfläche, um die komplette Auswahl aufzuheben.

Einen Block verschieben

Sie können jeden Block auf der Arbeitsfläche an eine andere Position verschieben. Wenn ein oder mehrere Verbindungen mit dem Block verknüpft sind, den Sie verschieben, dann werden die Verbindungen ebenfalls verschoben.

Einen Block verschieben

- Klicken Sie auf der Arbeitsfläche des Modelleditors auf den zu verschiebenden Block, halten Sie die Maustaste gedrückt und ziehen Sie den Block an die neue Position.
- Oder um mehrere Blöcke gleichzeitig zu verschieben, wählen Sie die Blöcke wie unter „Blöcke und Verbindungen auswählen“ auf Seite 72 beschrieben aus. Klicken und halten Sie dann die Maustaste gedrückt und ziehen Sie die Blöcke sofort an die neue Position (warten Sie nicht, bis sich der Mauszeiger verändert).

Einen Block duplizieren

Sie können jeden Block auf der Arbeitsfläche duplizieren. Der ursprüngliche Block und sein Duplikat haben dann dieselben Parameter.

Wenn Sie einen einzelnen Block duplizieren, dann werden die verknüpften Verbindungen nicht automatisch dupliziert. Wenn Sie jedoch mehrere Blöcke gleichzeitig duplizieren, dann werden die Verbindungen zwischen den ausgewählten Blöcken automatisch dupliziert.

Einen Block duplizieren

- Klicken Sie auf der Arbeitsfläche des Modelleditors auf den zu duplizierenden Block und führen Sie einen der folgenden Schritte aus:

- Klicken Sie auf den Befehl **Duplizieren**, der unten im Blockparametereditor angezeigt wird.
- Oder drücken Sie Strg+C um den Block zu kopieren, und drücken Sie anschließend Strg+V um den Block einzufügen.
- Oder drücken Sie die Strg-Taste und ziehen Sie den zu duplizierenden Block an die Position, an der Sie das Duplikat platzieren möchten.

- Oder um mehrere Blöcke gleichzeitig zu duplizieren, wählen Sie sie wie unter „Blöcke und Verbindungen auswählen“ auf Seite 72 beschrieben aus und gehen Sie dann wie oben beschrieben vor. Ausnahme: Der Befehl **Duplizieren** ist nur verfügbar, wenn Sie einen einzelnen Block auswählen.

Zwei Blöcke verbinden

Sie können die Blöcke auf der Arbeitsfläche mit den sogenannten *Verbindungen* verknüpfen. Dann wird die Ausgabe des einen Blocks als Eingabe für den anderen Blocks benutzt.

Die Verbindungen werden an die *Ports* angeschlossen. Die Ports sind die Kreise, die links und/oder rechts auf einem Block angezeigt werden. Jeder Block kann keinen, einen oder mehrere der folgenden Ports haben:

- Ausgangsports (auf der rechten Seite eines Blocks)
- Eingangsports (auf der linken Seite eines Blocks)

Die Namen der Ports werden angezeigt, wenn Sie auf den Block klicken und den Block somit auswählen. Sie können auch den Mauszeiger über einen Port bewegen, um den Portnamen in einem Tooltip anzuzeigen.

Ausführliche Informationen zu den Werttypen, die zwischen zwei Blöcken gesendet werden können, zur Verarbeitungsreihenfolge der Verbindungen, Einschränkungen und vieles mehr finden Sie unter „Verbindungen und Blöcke“ auf Seite 99.

Zwei Blöcke verbinden

- Klicken Sie auf der Arbeitsfläche des Modelleditors auf den Ausgangsport des Blocks, den Sie verbinden möchten, und ziehen Sie die Maus zum Eingangsport eines anderen Blocks.

Eine Verbindung ändern

Sie können den Weg ändern, den eine bestehende Verbindung zu dem aktuell verknüpften Block nimmt. Sie können eine Verbindung aber auch mit einem anderen Block oder einem anderen Port desselben Blocks verknüpfen.

Es ist nicht möglich, Kreisläufe mit den Verbindungen zu erzeugen. Ausführliche Informationen finden Sie unter „Verbindungseinschränkungen“ auf Seite 109.

Eine Verbindung ändern

1. Klicken Sie auf der Arbeitsfläche des Modelleditors auf die Verbindung, die Sie ändern möchten.

   Die Portnamen der verbundenen Blöcke werden angezeigt und die Ports an den beiden Enden der Verbindung werden optisch hervorgehoben.

2. Um den Weg zu ändern, den eine Verbindung zwischen zwei Blöcken nimmt, ziehen Sie eines der Größenänderungssymbole (■), die jetzt auf der markierten Verbindung angezeigt werden, an eine andere Position.

   Oder ziehen Sie das Verschiebesymbol (◆), das jetzt auf dem Ein- oder Ausgangsport angezeigt wird (in diesem Fall wird der Mauszeiger als Hand angezeigt) an einen anderen Port.

Einen Block oder eine Verbindung entfernen

Sie können jeden Block oder jede Verbindung entfernen, die sich zur Zeit auf der Arbeitsfläche gefindet. Wenn Sie einen Block entfernen, dann werden alle Verbindungen, die mit diesem Block verknüpft sind, automatisch entfernt.

Einen Block oder eine Verbindung entfernen

- Klicken Sie auf der Arbeitsfläche des Modelleditors auf den zu entfernenden Block oder die zu entfernende Verbindung und drücken Sie die Entf-Taste.

  Bei einem Block können Sie alternativ auf den Befehl **Entfernen** klicken, der unten im Blockparametereditor angezeigt wird.

- Oder wenn Sie mehrere Blöcke und/oder Verbindungen gleichzeitig entfernen möchten, wählen Sie sie aus wie unter „Blöcke und Verbindungen auswählen“ auf Seite 72 beschrieben und drücken Sie dann die Entf-Taste.

Eine Operation rückgängig machen und wiederherstellen

Sie können jede Änderung auf der Arbeitsfläche rückgängig machen und wiederherstellen. Sie können zum Beispiel das Entfernen von Blöcken, die Änderung von Parameterwerten oder den geänderten Weg einer Verbindung rückgängig machen.

Es ist nicht möglich, die Änderung eines Modellnamens oder der Modellbeschreibung mit den unten beschriebenen Schaltflächen rückgängig zu machen.

Anmerkung:
Die Arbeitsfläche muss den Fokus haben, damit die unten beschriebenen Tastenkombinationen funktionieren. Wenn die Dokumentationsanzeige oder die Palette aktuell den Fokus hat, dann wird die Änderung auf der Arbeitsfläche nicht rückgängig gemacht oder wiederhergestellt.

Eine Operation rückgängig machen oder wiederherstellen

- Um die letzte Operation rückgängig zu machen, klicken Sie in der Symbolleiste des Modelleditors auf .

Oder drücken Sie Strg+Z.

- Um die letzte Operation wiederherzustellen, klicken Sie in der Symbolleiste des Modelleditors auf .

  Oder drücken Sie Strg+Y.

Die oben genannten Schaltflächen sind nur dann in der Symbolleiste aktiviert, wenn eine Operation rückgängig gemacht oder wiederhergestellt werden kann.

Geräte, Gerätegruppen und Assets ersetzen

Sie können die Eingabe- und Ausgabeblöcke nach den Geräten, Gerätegruppen und Assets durchsuchen, die im aktuellen Modell verwendet werden, und sie durch andere Geräte, Gerätegruppen oder (wenn Sie das Suchfeld verwenden) Assets ersetzen, die zur Zeit im Cumulocity IoT-Inventar registriert sind (diese Geräte werden in der Device Management-Anwendung visualisiert).

Anmerkung:

In den Regeln unten bezieht sich der Begriff *Gerät* auf ein Gerät oder ein anderes Asset (jedoch nicht auf eine Gerätegruppe).

Es gelten die folgenden Regeln:

- Sie können ein Gerät durch ein anderes Gerät ersetzen.
- Sie können eine Gerätegruppe durch eine andere Gerätegruppe ersetzen.
- Wenn Sie ein Gerät durch eine Gerätegruppe ersetzen:
  - alle passenden Eingabegeräte werden zu Gerätegruppen abgeändert, und
  - alle passenden Ausgabegeräte werden in Trigger-Geräte abgeändert.
- Wenn Sie eine Gerätegruppe durch ein Gerät ersetzen:
  - eine Gerätegruppe wird in ein Gerät abgeändert, und
  - alle passenden Trigger-Geräte werden in das angegebene Gerät abgeändert.

    Anmerkung:

    Wenn Sie mehr als eine Gerätegruppe gleichzeitig in ein Gerät abändern, dann wird nur das erste angegebene Gerät zum Ersetzen aller Trigger-Geräte verwendet.
- Ein Eintrag mit dem Namen „Trigger-Gerät“ wird im Dialog nicht zur Auswahl angeboten.

Nach dem Gerätetausch müssen Sie sicherstellen, dass die von den Ein-und Ausgabeblöcken des aktuellen Modells benutzten Messwerte sich immer noch auf passende Messwerte beziehen. Das Cumulocity IoT-Fragment und die Serie werden durch den Austausch nicht geändert; diese Werte können womöglich auch für das neue Gerät gelten, das muss aber nicht so sein.

Geräte oder Gerätegruppen ersetzen

1. Klicken Sie in der Symbolleiste des Modelleditors auf . Diese Schaltfläche ist nur dann in der Symbolleiste aktiviert, wenn mindestens ein Gerät, eine Gerätegruppe oder ein Asset im aktuellen Modell definiert wurde. Definierte Trigger-Geräte werden hierbei nicht berücksichtigt.

2. Wählen Sie aus dem Dropdown-Listenfeld **Aktuelles Gerät und Gerätegruppe** des anschließend angezeigten Dialogs das Gerät, die Gerätegruppe oder das Asset, was ausgetauscht werden soll. Zur Auswahl stehen alle Geräte, Gerätegruppen und Assets, die im Modell verwendet werden.

3. Klicken Sie auf **Ersetzen durch** um ein Dialogfeld anzuzeigen. Das Dialogfeld ist dasselbe, das auch im Blockparametereditor bei der Auswahl eines anderen Geräts, einer anderen Gerätegruppe oder eines anderen Assets angezeigt wird. Weitere Informationen zu diesem Dialogfeld finden Sie unter „Die Parameter eines Blocks editieren“ auf Seite 69. Klicken Sie die Schaltfläche **Verwenden** (diese wird angezeigt, wenn Sie mit der Maus über einen Eintrag fahren), um das gewünschte Gerät, die Gerätegruppe oder (wenn das Suchfeld verwendet wurde) das Asset auszuwählen, das sie statt dessen verwenden möchten.

   Die maximale Anzahl der angezeigten Geräte, Gerätegruppen und/oder Assets wird durch die Mandantenoptionen bestimmt. Weitere Informationen finden Sie in „Die Anzahl der angezeigten Geräte, Gerätegruppen und/oder Assets konfigurieren“ auf Seite 141.

   Die Assets, die bei der Suche angezeigt werden, werden auch durch die Mandantenoptionen bestimmt. Sie können die Suche so einschränken, dass nur Assets eines bestimmten Typs angezeigt werden. Weitere Informationen finden Sie unter „Nach Eingabe- und Ausgabe-Assets suchen“ auf Seite 141.

4. Wenn Sie weitere Geräte, Gerätegruppen oder Assets austauschen möchten, klicken Sie auf **Zeile hinzufügen**. Dies wird nur angezeigt, wenn mehr als ein Gerät, eine Gerätegruppe oder ein Asset im aktuellen Modell definiert wurde.

   Eine neue Zeile mit zusätzlichen Dropdown-Listenfeldern für **Aktuelles Gerät und Gerätegruppe** und **Ersetzen durch** wird angezeigt. Sie können jetzt ein weiteres Gerät, eine weitere Gerätegruppe oder ein weiteres Asset für den Austausch auswählen. Die Geräte, Gerätegruppen und Assets, die Sie vorher bereits für den Austausch ausgewählt haben, werden im Dropdown-Listenfeld **Aktuelles Gerät und Gerätegruppe** nicht mehr zur Auswahl angeboten.

   Wiederholen Sie diesen Schritt so oft, bis Sie alle erforderlichen Geräte, Gerätegruppen und Assets für den Austausch ausgewählt haben. Sie können so viele Zeilen hinzufügen, wie sich Geräte, Gerätegruppen oder Assets im aktuellen Modell befinden.

5. Wenn Sie eine Zeile entfernen möchten (zum Beispiel weil Sie ein ausgewähltes Gerät doch nicht mehr austauschen möchten), klicken Sie auf das ✖ neben dieser Zeile. Dies steht nur dann zur Verfügung, wenn mehr als eine Zeile im Dialogfeld angezeigt wird.

6. Klicken Sie auf **Ersetzen**.

Vorlagenparameter verwalten

Ein Modell kann null oder mehr Vorlagenparameter enthalten, die bei den Blockparametern anstelle von definierten Werten verwendet werden können. Anstatt beim **Schwelle**-Block zum Beispiel einen Schwellenwert von 100 zu definieren, können Sie im Blockparametereditor einen Vorlagenparameter zuweisen.

Wenn ein Modell Vorlagenparameter hat, dann können mehrere Instanzen des Modells erstellt werden, und es können unterschiedliche Werte für die Vorlagenparameter angegeben werden. Siehe auch „Modelle“ auf Seite 40; dort wird der Unterschied zwischen Modellen ohne Vorlagenparameter und Modellen mit Vorlagenparametern erläutert, sowie die verschiedenen Rollen: Modellautor und Instanzenverwalter.

Wenn Sie Vorlagenparameter verwenden möchten, müssen Sie diese zuerst, wie unten beschrieben, im Dialogfeld "Vorlagenparameter" definieren. Nachdem Sie einen Vorlagenparameter definiert haben, können Sie ihn im Blockparametereditor bestimmten Parametern desselben Typs zuweisen (siehe auch „Die Parameter eines Blocks editieren“ auf Seite 69).

Anmerkung:

Sie können die Vorlagenparameter auch direkt im Blockparametereditor definieren.

Modelle ohne Vorlagenparameter können direkt im Modellmanager aktiviert werden. Hierbei wird immer nur eine einzelne Modellinstanz ausgeführt. Sobald Sie einem oder mehreren Blockparametern mindestens einen Vorlagenparameter zuweisen und das Modell speichern, ändert sich dieses Verhalten. Sie können das Modell dann nicht mehr direkt im Modellmanager aktivieren. Stattdessen müssen Sie mindestens eine Instanz des Modells erstellen, und diese Instanz können Sie dann im Instanzeditor aktivieren. Ausführliche Informationen finden Sie unter „Mit dem Instanzeditor arbeiten“ auf Seite 91.

Vorlagenparameter für die Instanzen des aktuellen Modells definieren

1. Klicken Sie in der Symbolleiste des Modelleditors auf [x] um das Dialogfeld "Vorlagenparameter" aufzurufen.

   Auf dem Symbol wird ein Häkchen angezeigt, wenn mindestens ein Vorlagenparameter definiert wurde: [x].

   Das Dialogfeld ist anfangs leer und Sie müssen die Vorlagenparameter erstellen, die Sie in Ihrem Modell verwenden möchten. Wenn bereits Vorlagenparameter definiert wurden, dann werden diese im Dialogfeld angezeigt.

   Wenn es eine lange Liste von Vorlagenparametern gibt, dann können Sie den gesuchten Vorlagenparameter leicht finden, indem Sie seinen Namen oder einen Teil des Namens im Suchfeld eingeben. Sobald Suchkriterien angewendet werden, wird ✖ im Suchfeld angezeigt. Klicken Sie darauf, wenn Sie die Suchkriterien löschen und alle verfügbaren Vorlagenparameter wieder anzeigen möchten.

Nachdem einem oder mehreren Blockparametern ein Vorlagenparameter zugewiesen wurde, wird in der Spalte **Verwendung** die Anzahl der Stellen angezeigt, an denen der Vorlagenparameter verwendet wird. Wenn ein Vorlagenparameter zum Beispiel an zwei Stellen verwendet wird, so bedeutet dies, dass er zwei Blockparametern zugewiesen wurde. Diese können sich im selben Block oder in verschiedenen Blöcken befinden.

2. Klicken Sie auf **Neuer Vorlagenparameter**, um einen neuen Vorlagenparameter zu erstellen.

   Hierdurch wird am unteren Ende des Dialogfelds eine leere Zeile hinzugefügt. Sie können **Neuer Vorlagenparameter** mehrere Male anklicken, um mehrere leere Zeilen hinzuzufügen, die Sie nacheinander ausfüllen können.

3. Geben Sie für jeden Vorlagenparameter die folgenden Informationen an:

   - **Name**. Geben Sie einen eindeutigen Namen ein, der den Vorlagenparameter im aktuellen Modell identifiziert. Dieser Name kann später im Blockparametereditor ausgewählt werden.
   - **Typ**. Wählen Sie aus dem Dropdown-Listenfeld den Wertetyp des Vorlagenparameters aus. Der Typ kann zum Beispiel ein String- oder Float-Wert sein, der Name eines Geräts oder einer Gerätegruppe, oder der Parameter eines bestimmten Blocks.

     Ein Eingabeblock bezieht sich auf ein Gerät oder eine Gerätegruppe, während sich ein Ausgabeblock auf ein Gerät oder ein Trigger-Gerät bezieht. Derselbe Vorlagenparameter und damit derselbe Wert kann sowohl bei einem Eingabeblock als auch bei einem Ausgabeblock verwendet werden. Wenn sich ein Vorlagenparameter auf eine Gerätegruppe bezieht, dann wird er bei einem Ausgabeblock als Trigger-Gerät behandelt. In der Regel wird ein einzelner Vorlagenparameter für alle Eingabe- und Ausgabeblöcke verwendet und kann für ein einzelnes Gerät oder eine Gerätegruppe stehen. In diesem Fall geht die Ausgabe des Blocks an das Gerät innerhalb der Gruppe, die eine Modellauswertung ausgelöst hat (ein Modell, das den Durchschnitt einer Messung berechnet und an eine Messung ausgibt, würde somit unabhängig für jedes Gerät eine neue Messung erzeugen). Auch wenn man einen anderen Vorlagenparameter verwenden würde, dessen Wert sich auf eine andere Gruppe bezieht, würde die Ausgabe des Modells nur an das Gerät gesendet werden, welches die Auswertung des Modells ausgelöst hat.
   - **Optional**. Ein optionaler Wert kann leer bleiben oder später vom Instanzenverwalter festgelegt werden. Wenn Sie dieses Kontrollkästchen aktivieren, dann ist es nicht möglich, einem Standardwert anzugeben.
   - **Standardwert**. Sie können einen Standardwert nur dann angeben, wenn das Kontrollkästchen **Optional** nicht aktiviert ist.

     Ausnahme: Boolesche Typen haben immer einen Wert und können nicht optional sein. Sie sind standardmäßig „falsch“ (das heißt, das Kontrollkästchen für den Standardwert ist nicht aktiviert).

     Wenn Sie einen Standardwert angeben, dann wird dieser Standardwert im Instanzeditor angezeigt, wenn der Instanzenverwalter eine neue Instanz erstellt. Der Instanzenverwalter kann diesen Standardwert dann entweder unverändert lassen oder für diese Instanz nach Bedarf ändern.

Wenn Sie den Standardwert für ein Gerät festlegen und hierfür das Feld **Standardwert** anklicken, dann wird ein zusätzliches Dialogfeld angezeigt. Dieses Dialogfeld ist dasselbe wie bei der Auswahl eines anderen Geräts, einer anderen Gerätegruppe oder eines anderen Assets im Blockparametereditor (weitere Informationen zu diesem Dialogfeld finden Sie unter „Die Parameter eines Blocks editieren“ auf Seite 69). Klicken Sie die Schaltfläche **Verwenden** (diese wird angezeigt, wenn Sie mit der Maus über einen Eintrag fahren), um das gewünschte Gerät auszuwählen.

Anmerkung:

Wenn es einen Blockparameter gibt, bei dem kein erforderlicher Wert angegeben wurde, dann kann die Instanz nicht aktiviert werden. Bei einem Aktivierungsversuch wird ein Fehler angezeigt.

4. Sie können einen Vorlagenparameter jederzeit aktualisieren. Dies beinhaltet den Namen, ob er optional ist oder nicht, sowie den Standardwert. Alle Blöcke, in denen der aktualisierte Vorlagenparameter definiert ist, werden automatisch an die Verwendung der neuen Werte angepasst. Die einzige Ausnahme ist der Typ. Sie können den Typ nur ändern, wenn der Vorlagenparameter in keinem Block des Modells verwendet wird.

5. Wenn das Modell inaktiv ist, können Sie die Vorlagenparameter neu anordnen. Dies beeinflusst die Reihenfolge, in der sie im Instanzeditor angezeigt werden. Ziehen Sie eine Zeile mit Hilfe von ⇕ neben dieser Zeile an eine andere Position. Siehe auch „Die Instanzen filtern und sortieren“ auf Seite 95.

6. Sie können einen Vorlagenparameter nur entfernen, wenn er in keinem Block des Modells verwendet wird. Um einen Vorlagenparameter zu entfernen, klicken Sie auf das Aktionsmenü (die drei vertikalen Punkte am Ende einer Zeile) und klicken Sie dann auf **Entfernen**.

7. Klicken Sie auf **OK**, um die Änderungen im Datenspeicher zu behalten und das Dialogfeld zu schließen.

Anmerkung:

Denken Sie daran, dass Ihre Änderungen nur dann dauerhaft ins Inventar geschrieben werden, wenn Sie das Modell speichern. Siehe auch „Ein Modell speichern“ auf Seite 66.

Elemente in ein anderes Modell kopieren

Sie können alle beliebigen Elemente auf der Arbeitsfläche (Blöcke, Gruppen und verknüpfte Verbindungen) kopieren und in ein anderes Modell einfügen. Die Voraussetzung hierfür ist, dass alles in derselben Sitzung gemacht wird. Es funktioniert nicht, wenn Sie versuchen, die Elemente auf einer anderen Registerkarte oder in einem anderen Browser einzufügen.

ACHTUNG:

Wenn Sie viele Ein- und Ausgabeblöcke kopieren, können Performance-Probleme auftreten weil diese Operation Zugriff auf den Inventardienst von Cumulocity IoT benötigt, um Informationen zu den Geräten zu erhalten, die von diesen Blöcke repräsentiert werden.

Elemente in ein anderes Modell kopieren

1. Wählen Sie auf der Arbeitsfläche des Modelleditors alle Elemente aus, die Sie kopieren möchten und drücken Sie Strg+C.

   Dies funktioniert auch, wenn sich das Modell im Lesemodus befindet.

2. Verlassen Sie den Modelleditor. Siehe auch „Den Modelleditor verlassen“ auf Seite 67.

3. Wechseln Sie im Modellmanager zu dem Modell, in das Sie die kopierten Elemente einfügen möchten. Dies kann ein bestehendes Modell sein (siehe auch „Ein bestehendes Modell bearbeiten“ auf Seite 55) oder ein neues Modell, das Sie zuerst erstellen müssen (siehe auch „Ein neues Modell erstellen“ auf Seite 54).

4. Wenn der Modelleditor angezeigt wird, drücken Sie Strg+V, um die kopierten Elemente in das Modell einzufügen.

Mit Gruppen arbeiten

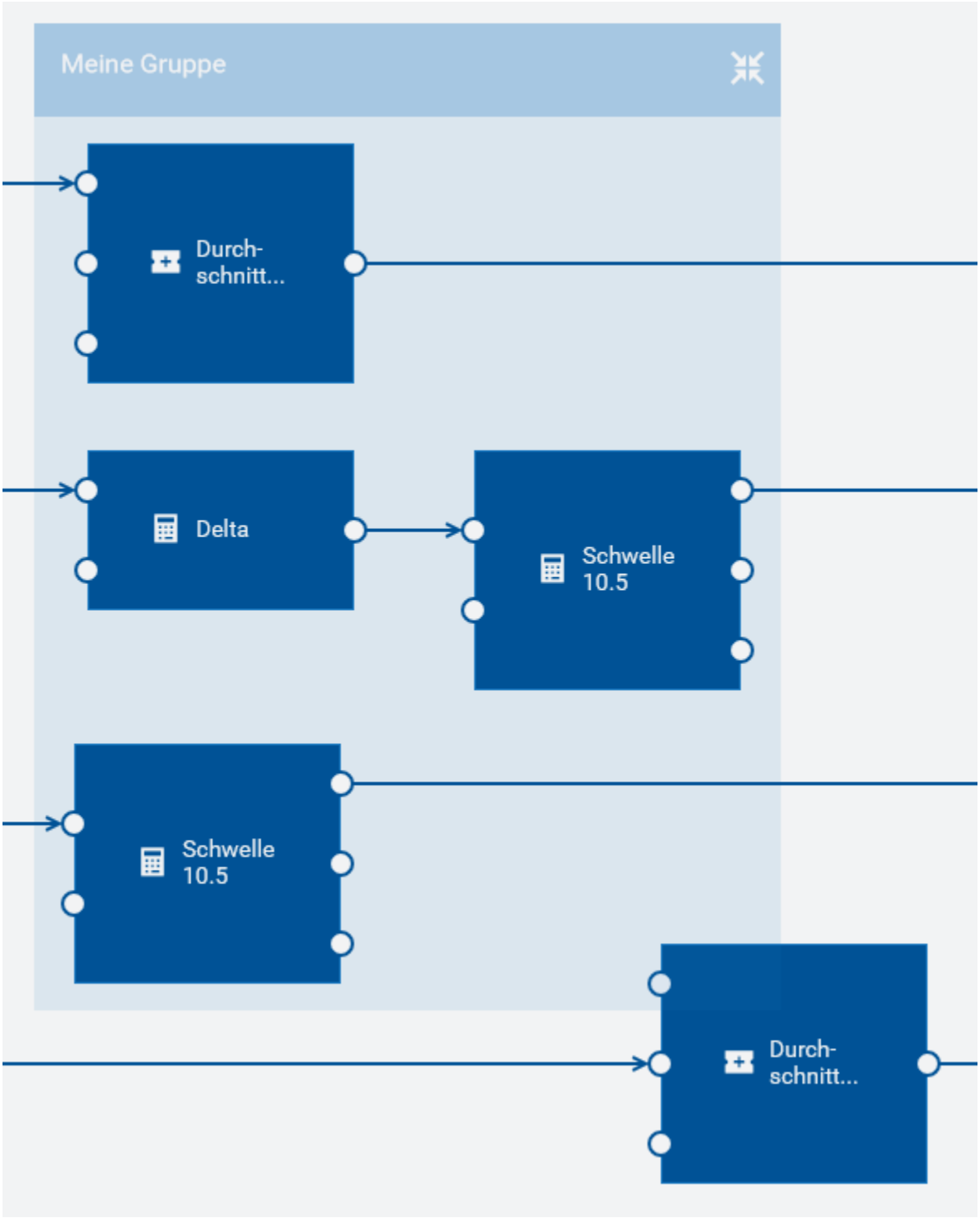
Was ist eine Gruppe?

Sie können Blöcke und die zugehörigen Verbindungen in einer Gruppe anordnen. Eine Gruppe ist ein spezieller Blocktyp, der auf- und zugeklappt werden kann. Wenn eine Gruppe aufgeklappt ist, können Sie ihren Inhalt genauso bearbeiten wie auf der Arbeitsfläche. Sie können zum Beispiel Verbindungen hinzufügen oder die Blockparameter ändern. Sie können der Gruppe auch weitere Blöcke hinzufügen oder Blöcke aus der Gruppe entfernen. Wenn eine Gruppe zusammengeklappt ist, nimmt sie weniger Platz auf der Arbeitsfläche ein; die Blöcke und die Verbindungen innerhalb der Gruppe sind in diesem Fall jedoch nicht zu sehen.

Gruppen sind hilfreich, wenn häufig benötigte Funktionen an mehreren Stellen verfügbar gemacht werden sollen. Sie können jeder Gruppe einen Namen geben, mit dem sie identifiziert werden kann. Sie können eine Gruppe kopieren und entweder in dasselbe Modell oder in ein anderes Modell einfügen.

Anmerkung:

Verwechseln Sie diese Art von Gruppe nicht mit einer Gerätegruppe. Eine Gerätegruppe ist ein spezieller Eingabeblock, den Sie aus der Palette auswählen können. Siehe auch „Einen Block hinzufügen“ auf Seite 67.



Die Größe der Box, die für eine Gruppe angezeigt wird, wird durch ihren Inhalt bestimmt. Wenn Sie einen Block innerhalb der Gruppe an eine andere Position verschieben, wird die Größe der Box automatisch angepasst (das heißt, die Box wird größer oder kleiner). Dasselbe gilt, wenn Sie den Weg ändern, den eine Verbindung zu einem anderen Block innerhalb derselben Gruppe nimmt.

Sie verschieben eine Gruppe auf der Arbeitsfläche genauso wie Sie einen Block verschieben (siehe auch „Einen Block verschieben“ auf Seite 73). Wenn Sie eine Gruppe verschieben, dann wird die Gruppe immer über allen anderen Elementen auf der Arbeitsfläche angezeigt. Die Box, die für eine Gruppe angezeigt wird, ist transparent; so können Sie leicht erkennen, welche Blöcke zu der Gruppe gehören und welche nur von der Box überlagert werden.

Es ist nicht möglich, Gruppen zu verschachteln.

Anmerkung:

Es gibt eine Ausnahme beim Bearbeiten des Inhalts einer Gruppe: Wenn Sie einen oder mehrere Blöcke in der Gruppe mit Strg+C und Strg+V duplizieren oder wenn Sie im Blockparametereditor den Befehl **Duplizieren** verwenden, dann wird das Duplikat nicht zur Gruppe hinzugefügt. Sie wird statt dessen direkt auf der Arbeitfläche abgelegt. Wenn Sie jedoch die Strg-Taste drücken und die zu duplizierenden Blöcke mit der Maus ziehen, dann können Sie das Duplikat entweder innerhalb der Gruppe (dies kann dieselbe oder eine andere Gruppe sein) oder auf der Arbeitsfläche platzieren. Siehe auch „Einen Block duplizieren“ auf Seite 73.

Eine Gruppe erstellen

Sie können jeden beliebigen auf der Arbeitsfläche angezeigten Block (einschließlich der Verbindungen zwischen den Blöcken) zu einer Gruppe hinzufügen.

Es ist nicht möglich, eine leere Gruppe zu erstellen. Sie müssen eine Gruppe zuerst wie unten beschrieben erstellen. Sobald die Gruppe vorhanden ist, können Sie der Gruppe weitere Blöcke hinzufügen, entweder von der Palette oder der Arbeitsfläche (siehe auch „Einen Block hinzufügen“ auf Seite 67 und „Blöcke in eine Gruppe verschieben“ auf Seite 85).

Eine Gruppe erstellen

1. Wählen Sie auf der Arbeitsfläche des Modelleditors einen oder mehrere Blöcke aus, die Sie einer Gruppe hinzufügen möchten. Die Verbindungen brauchen Sie nicht auswählen; alle bestehenden Verbindungen bleiben erhalten. Siehe auch „Blöcke und Verbindungen auswählen“ auf Seite 72.

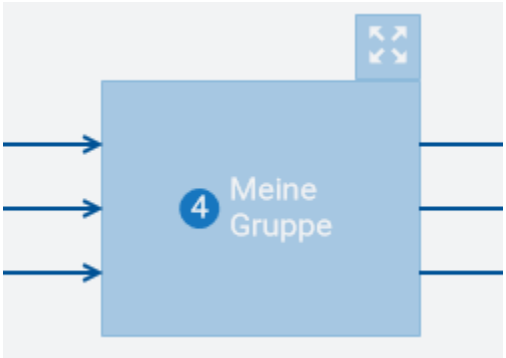
2. Klicken Sie in der Symbolleiste des Modelleditors auf .

   Oder drücken Sie Strg+G.

Eine Gruppe auf- und zuklappen

Wenn Sie mehr Platz auf der Arbeitsfläche benötigen und der Inhalt der Gruppe nicht sichtbar sein muss, können Sie die Gruppe zuklappen.

Wenn eine Gruppe zugeklappt ist, wird in eine Zahl auf der Gruppe angezeigt. Diese gibt die Anzahl der Blöcke in dieser Gruppe an. Zum Beispiel:



Wenn Sie den Inhalt der Gruppe wieder sichtbar machen möchten (zum Beispiel, um die Blockparameter zu bearbeiten oder neue Verbindungen hinzuzufügen), müssen Sie die Gruppe wieder aufklappen.

Wenn Sie das Modell speichern, dann wird auch der Status jeder Gruppe (das heißt, ob sie zur Zeit auf- oder zugeklappt ist) gespeichert. Wenn Sie das Modell danach wieder bearbeiten, dann wird der Inhalt so wie beim letzten Speichern angezeigt.

Eine Gruppe auf- oder zuklappen

- Um eine Gruppe zuzuklappen, klicken Sie auf neben dem Gruppennamen.
- Um eine Gruppe aufzuklappen, klicken Sie auf rechts oben über der zugeklappten Gruppe.

Eine Gruppe umbenennen

Wenn Sie eine Gruppe erstellen, dann ist „Gruppe“ der Standardname. Sie können jede Gruppe umbenennen und ihr einen eindeutigen Namen geben.

Wenn ein Gruppenname länger ist als auf der Gruppe angezeigt werden kann, dann können Sie den Mauszeiger über den Gruppennamen bewegen, um den kompletten Namen in einem Tooltip anzuzeigen.

Es kann nicht passieren, dass eine Gruppe keinen Namen hat. Wenn Sie einen Gruppennamen löschen, dann wird der vorherige Name automatisch wieder verwendet.

Eine Gruppe umbenennen

1. Wählen Sie im Modelleditor die Gruppe aus und klicken Sie auf den Gruppennamen. Es spielt keine Rolle, ob die Gruppe derzeit auf- oder zugeklappt ist (siehe auch „Eine Gruppe auf- und zuklappen“ auf Seite 83). Dadurch wird der gesamte Name zur Bearbeitung ausgewählt.

2. Geben Sie einen neuen Gruppennamen an und drücken Sie die Eingabetaste.

Blöcke in eine Gruppe verschieben

Sie können einen oder mehrere Blöcke von der Arbeitsfläche in eine vorhandene Gruppe verschieben. Alle vorhandenen Verbindungen bleiben erhalten.

Sie können auch einen Block aus der Palette in eine vorhandene Gruppe ziehen. Siehe „Einen Block hinzufügen“ auf Seite 67.

Sie können die Blöcke nur dann in eine Gruppe verschieben/ziehen, wenn der Inhalt der Gruppe sichtbar ist, das heißt, wenn die Gruppe aufgeklappt ist. Siehe auch „Eine Gruppe auf- und zuklappen“ auf Seite 83.

Blöcke in eine Gruppe verschieben

1. Stellen Sie sicher, dass die Gruppe, in die Sie die Blöcke verschieben möchten, nicht zugeklappt ist.

2. Wählen Sie auf der Arbeitsfläche des Modelleditors die Blöcke aus, die Sie in die Gruppe verschieben möchten (siehe auch „Blöcke und Verbindungen auswählen“ auf Seite 72). Sie brauchen die Verbindungen zwischen den Blöcken nicht auswählen; diese werden automatisch zusammen mit den Blöcken verschoben.

3. Führen Sie einen der folgenden Schritte aus:

   - Ziehen Sie die Auswahl in die Gruppe und legen Sie sie dort ab.
   - Oder wählen Sie die Gruppe aus, in die Sie die Blöcke verschieben möchten. Klicken Sie in der Symbolleiste des Modelleditors anschließend auf oder drücken Sie Strg+G.

Blöcke aus einer Gruppe auf die Arbeitsfläche verschieben

Sie können einen Block aus einer Gruppe auf die Arbeitsfläche verschieben. Alle vorhandenen Verbindungen bleiben erhalten.

Eine Gruppe wird automatisch entfernt, wenn das letzte Element aus der Gruppe auf die Arbeitsfläche verschoben wurde. Wenn Sie alle Elemente gleichzeitig auf die Arbeitsfläche verschieben möchten, können Sie die Gruppierung einfach aufheben. Siehe „Die Gruppierung einer Gruppe aufheben“ auf Seite 86.

Blöcke aus einer Gruppe auf die Arbeitsfläche verschieben

- Um einen oder mehrere Blöcke gleichzeitig zu verschieben:

  1. Wählen Sie in der aufgeklappten Gruppe alle Blöcke, die Sie verschieben möchten.

  2. Klicken Sie in der Symbolleiste des Modelleditors auf .

     Oder drücken Sie Strg+Umschalt+G.

- Oder um einen einzelnen Block zu verschieben:

  1. Wählen Sie in der aufgeklappten Gruppe den Block, den Sie verschieben möchten.
  2. Klicken Sie unten im unten im Blockparametereditor auf den Befehl **Gruppierung aufheben**.

Blöcke und Verbindungen aus einer Gruppe entfernen

Sie entfernen die Blöcke und Verbindungen aus einer Gruppe genauso wie Sie sie direkt von der Arbeitsfläche entfernen. Die einzige Voraussetzung ist, dass die Gruppe aufgeklappt ist. Siehe „Einen Block oder eine Verbindung entfernen“ auf Seite 75.

Wenn das letzte Element aus einer Gruppe entfernt wurde, dann wird die Gruppe automatisch entfernt.

Eine Gruppe duplizieren

Sie können jede Gruppe auf der Arbeitsfläche duplizieren. Die ursprüngliche Gruppe und das Duplikat haben dann denselben Inhalt.

Verbindungen, die von Blöcken außerhalb der Gruppe kommen oder die von der Gruppe zu Blöcken außerhalb der Gruppe gehen, werden nicht dupliziert.

Sie können eine Gruppe auch in ein anderes Modell kopieren, siehe „Elemente in ein anderes Modell kopieren“ auf Seite 80.

Eine Gruppe duplizieren

- Klicken Sie auf der Arbeitsfläche des Modelleditors auf die Gruppe, die Sie duplizieren möchten (es spielt keine Rolle, ob die Gruppe derzeit auf- oder zugeklappt ist), und führen Sie einen der folgenden Schritte aus:
  - Drücken Sie Strg+C, um die Gruppe zu kopieren, und drücken Sie dann Strg+V, um die Gruppe einzufügen.
  - Oder drücken Sie Strg und ziehen Sie die zu duplizierende Gruppe an die Position, an der Sie das Duplikat platzieren möchten.

Die Gruppierung einer Gruppe aufheben

Wenn Sie die Gruppierung einer Gruppe aufheben, dann wird die Gruppe gelöscht und alle Blöcke aus dieser Gruppe werden auf die Arbeitsfläche verschoben. Alle vorhandenen Verbindungen bleiben erhalten.

Sie können die Gruppierung für mehrere Gruppen gleichzeitig aufheben. In diesem Fall ist es wichtig, dass innerhalb oder außerhalb der ausgewählten Gruppen kein Block und keine Verbindung ausgewählt ist. Andernfalls ist das Aufheben der Gruppierung nicht möglich.

Die Gruppierung einer Gruppe aufheben

1. Wählen Sie auf der Arbeitsfläche des Modelleditors eine oder mehrere Gruppen aus, deren Gruppierung Sie aufheben möchten. Dabei spielt es keine Rolle, ob eine Gruppe derzeit auf- oder zugeklappt ist.

2. Klicken Sie in der Symbolleiste des Modelleditors auf .

   Oder drücken Sie Strg+Umschalt+G.

Eine Gruppe entfernen

Sie können jede Gruppe von der Arbeitsfläche entfernen. Verbindungen zu Blöcken außerhalb der Gruppe werden entfernt.

ACHTUNG:
Wenn Sie eine Gruppe entfernen, dann werden alle Blöcke und Verbindungen in dieser Gruppe aus dem Modell entfernt.

Eine Gruppe entfernen

- Klicken Sie auf der Arbeitsfläche des Modelleditors auf die Gruppe, die Sie entfernen möchten (es spielt keine Rolle, ob die Gruppe derzeit auf- oder zugeklappt ist), und drücken Sie die Entf-Taste.

Die Arbeitsfläche verwalten

In einem großen Modell navigieren

Wenn Ihr Modell sehr groß ist und nicht alles im sichtbaren Bereich der Arbeitsfläche angezeigt werden kann, dann können Sie die Teile des Modells, die sich zur Zeit außerhalb des Fensters befinden, mit der Maus in den sichtbaren Bereich ziehen. Sie können dies entweder direkt auf der Arbeitsfläche oder in der Modellübersicht tun. Die Modellübersicht zeigt immer das gesamte Modell. Wenn die Modellübersicht derzeit nicht angezeigt wird, siehe „Die Modellübersicht anzeigen und ausblenden“ auf Seite 88.

In einem großen Modell navigieren

1. Positionieren Sie die Maus im Modelleditor über einer freien Stelle der Arbeitsfläche (eine Stelle, die keinen Block und keine Verbindung enthält) oder über einer beliebigen Stelle in der Modellübersicht.

2. Klicken und halten Sie die Maustaste gedrückt und ziehen Sie die Maus sofort in die gewünschte Richtung. Lassen Sie die Maustaste los, wenn der gewünschte Bereich auf der Arbeitsfläche zu sehen ist.

   Anmerkung:

Wenn Sie die Maustaste länger über einer freien Stelle der Arbeitsfläche gedrückt halten, ändert sich der Mauszeiger und Sie können stattdessen einen Bereich auswählen (zum Beispiel mehrere Blöcke und verknüpfte Verbindungen). Siehe auch „Blöcke und Verbindungen auswählen“ auf Seite 72.

Die Modellübersicht anzeigen und ausblenden

Die Modellübersicht zeigt das gesamte Modell. Sie wird rechts unten auf der Arbeitsfläche angezeigt. Wenn Sie die Modellübersicht nicht benötigen, können Sie sie ausblenden.

Die Modellübersicht anzeigen oder ausblenden

- Um die Modellübersicht zu verbergen, klicken Sie auf (direkt über der Modellübersicht).
- Um die Modellübersicht anzuzeigen, klicken Sie auf (rechts unten auf der Arbeitsfläche).

Die Arbeitsfläche zoomen

Die Symbolleiste des Modelleditors zeigt den aktuellen Zoom-Prozentsatz für die Arbeitsfläche an. Mit den Zoom-Schaltflächen in der Symbolleiste können Sie die Arbeitsfläche

- vergrößern, wodurch alles auf der Arbeitsfläche kleiner wird und mehr Elemente angezeigt werden können;
- verkleinern, wodurch alles auf der Arbeitsfläche größer wird, aber weniger Elemente angezeigt werden können.

Anmerkung:
Wenn Sie die unten aufgeführten Tastenkombinationen verwenden, dann bestimmt der zur Zeit aktive Bereich, was gezoomt werden soll. Wenn die Arbeitsfläche den Fokus hat (zum Beispiel wenn Sie gerade einen Block oder eine Verbindung ausgewählt haben), dann wird nur der Inhalt der Arbeitsfläche gezoomt. Wenn die Dokumentationsanzeige oder die Palette den Fokus hat, dann wird die Zoom-Funktion des Browsers verwendet und der gesamte Browserinhalt wird gezoomt (und der Zoom-Prozentsatz in der Symbolleiste bleibt unverändert).

Den Arbeitsbereich zoomen

- Zum Verkleinern klicken Sie in der Symbolleiste des Modelleditors auf .

  Oder drücken Sie Strg und die Minustaste.

- Zum Vergrößern klicken Sie in der Symbolleiste des Modelleditors auf .

  Oder drücken Sie Strg und die Plustaste.

Das Raster anzeigen und ausblenden

Die Blöcke, Verbindungen und Gruppen auf der Arbeitsfläche werden immer an einem Raster ausgerichtet. Sie können entscheiden, ob das Raster angezeigt werden soll oder nicht. Das Raster wird standardmäßig nicht angezeigt. Wenn Sie die Arbeitsfläche zoomen, dann wird das Raster entsprechend mitgezoomt.

Das Raster anzeigen oder ausblenden

- Klicken Sie in der Symbolleiste des Modelleditors auf die Schaltfläche zum Umschalten der Ansicht des Rasters.

  Wenn das Raster ausgeblendet ist, dann sieht die Schaltfläche so aus: .

  Wenn das Raster angezeigt wird, dann sieht die Schaltfläche so aus: .

  Wenn sich das Modell im Status "Aktiv" befindet (Lesemodus), dann ist es nicht möglich, die Anzeige des Rasters umzuschalten und diese Schaltfläche ist daher deaktiviert.

6 Mit dem Instanzeditor arbeiten

- Die Benutzeroberfläche des Instanzeditors 92
- Eine Instanz erstellen 93
- Eine Instanz editieren 93
- Eine Instanz aktivieren 94
- Eine Instanz deaktivieren 95
- Die Instanzen filtern und sortieren 95
- Eine Instanz duplizieren 96
- Eine Instanz entfernen 96
- Die Instanzen speichern 96
- Die Instanzen neu laden 97
- Den Instanzeditor verlassen 97

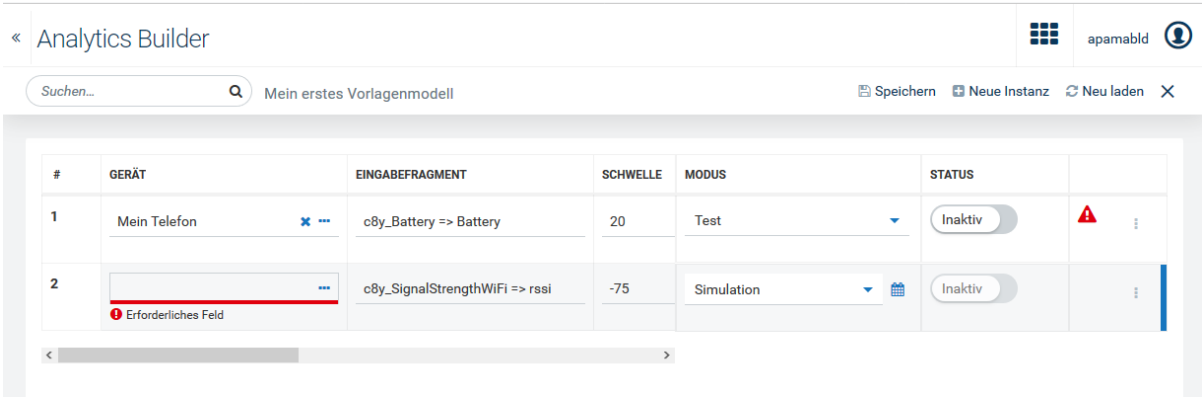
Die Benutzeroberfläche des Instanzeditors

Eine Voraussetzung für den Aufruf des Instanzeditors ist, dass ein oder mehrere Vorlagenparameter im Modelleditor definiert wurden (siehe auch „Vorlagenparameter verwalten“ auf Seite 78).

Mit dem Instanzeditor können Sie verschiedene Instanzen desselben Modells erstellen. Die Blöcke in jeder Instanz können dann unterschiedliche Werte für die Vorlagenparameter verwenden, und die Instanzen können unabhängig von den anderen Instanzen aktiviert werden. Sie rufen den Instanzeditor über den Modellmanager auf. Siehe auch „Die Instanzen eines Modells bearbeiten“ auf Seite 56.

Anmerkung:
Um die Instanzen zu editieren, benötigen Sie die ADMIN-Berechtigung für "Regeln" (englisch: CEP management). Wenn Sie über die LESEN-Berechtigung verfügen, können Sie die Instanzen nur anzeigen.

Der Instanzeditor zeigt die Instanzen des ausgewählten Modells an. Wenn Instanzen vorhanden sind, dann wird in einer Tabelle folgendes angezeigt: die Werte der Instanzen, der Modus und ob die Instanz aktiv ist.



Für jede Instanz wird eine Zeile angezeigt. Es gibt eine Spalte für jeden im Vorlagenmodell definierten Vorlagenparameter, wobei der Name des Vorlagenparameters die Spaltenüberschrift ist. Wenn eine Instanz nicht aktiv ist, dann können Sie die Werte für diese Instanz ändern.

Eine horizontale Bildlaufleiste ist verfügbar, wenn nicht alle Vorlagenparameter (Spalten) auf dem Bildschirm angezeigt werden können.

Die rechte Seite der Tabelle zeigt den Modus und Status jeder Instanz. Hier aktivieren und deaktivieren Sie die Instanzen. Siehe auch „Eine Instanz aktivieren“ auf Seite 94 und „Eine Instanz deaktivieren“ auf Seite 95.

Jede Zeile, die für eine Instanz angezeigt wird, hat ein Aktionsmenü (die drei vertikalen Punkte ganz rechts in der Zeile) mit Befehlen zum Verwalten der Instanz (zum Beispiel, zum Entfernen der Instanz).

Sie können die Liste der Instanzen filtern und sortieren. Weitere Informationen finden Sie unter „Die Instanzen filtern und sortieren“ auf Seite 95.

Wenn das Fehlersymbol am Ende einer Zeile angezeigt wird, dann verarbeitet die entsprechende Instanz keine Ereignisse mehr. Klicken Sie auf dieses Symbol, um weitere Informationen zu erhalten.

Wenn Sie den Instanzeditor öffnen, kann es passieren, dass die Vorlagenparameter seit der letzten Bearbeitung der Instanzen geändert wurden und nicht mehr dieselben Wertetypen wie zuvor verwenden. Wenn die im Instanzeditor angegebenen Werte noch kompatibel sind, werden sie in die neuen Wertetypen konvertiert. Inkompatible Werte (einschließlich der Kontrollkästchen für boolesche Typen und in Dropdown-Listenfeldern angezeigte Werte) werden automatisch entfernt. Jedes Feld, aus dem der Wert entfernt wurde, erhält eine Fehlerunterstreichung mit einer entsprechenden Fehlermeldung.

Eine Instanz erstellen

Wenn Sie eine neue Instanz erstellen, wird der Tabelle des Instanzeditors eine neue Zeile hinzugefügt. Sie können die erforderlichen Werte dann entweder sofort eingeben oder zuerst alle erforderlichen Zeilen hinzufügen und die Zeilen dann nacheinander füllen.

Eine Instanz erstellen

1. Klicken Sie in der Symbolleiste des Instanzeditors auf **Neue Instanz**.

   Dadurch wird am Ende der Tabelle eine neue Zeile hinzugefügt. Neue Instanzen (Zeilen) werden mit einer Hintergrundfarbe angezeigt, bis sie gespeichert werden.

2. Geben Sie die Werte für die im Modell definierten Vorlagenparameter ein. Siehe auch „Eine Instanz editieren“ auf Seite 93.

Eine Instanz editieren

Sie geben die Parameterwerte für die Instanzen genauso an wie die Werte für die Blöcke im Modelleditor (siehe auch „Die Parameter eines Blocks editieren“ auf Seite 69).

Die Tabelle im Instanzeditor bietet, in Abhängigkeit vom Typ des Vorlagenparameters, verschiedene Arten von Steuerelementen für die Eingabe:

- Es gibt Textfelder, in die Sie je nach Einstellung des Vorlagenparameters Werte eingeben können (zum Beispiel, einen String- oder Float-Wert). Die Werte werden während der Eingabe überprüft. Es ist zum Beispiel nicht möglich, enen String-Wert in einem Textfeld einzugeben, das einen Float-Wert erwartet.
- Es gibt Kontrollkästchen für boolesche Werte. Wenn ein Kontrollkästchen aktiviert wird, dann wird der Wert auf `true` gesetzt.
- Es gibt Dropdown-Listenfelder aus denen Sie einen anderen Wert auswählen können (zum Beispiel um eine andere Regel für das Runden auszuwählen).
- Ein zusätzliches Dialogfeld erscheint, wenn Sie ein Gerät oder eine Gerätegruppe editieren. Dieses Dialogfeld ist dasselbe wie bei der Auswahl eines anderen Geräts, einer anderen

Gerätegruppe oder eines anderen Assets im Blockparametereditor (weitere Informationen zu diesem Dialogfeld finden Sie unter „Die Parameter eines Blocks editieren“ auf Seite 69). Klicken Sie die Schaltfläche **Verwenden** (diese wird angezeigt, wenn Sie mit der Maus über einen Eintrag fahren), um das gewünschte Gerät auszuwählen.

Editierte Instanzen (Zeilen), die noch nicht gespeichert wurden, werden mit einer Hintergrundfarbe angezeigt, bis sie gespeichert werden.

Wenn ein Textfeld einen Wert benötigt ist, der noch nicht angegeben wurde, dann wird eine Meldung angezeigt, die darauf hinweist, dass dies ein erforderliches Feld ist. Es ist möglich, die Instanzen zu speichern, den Instanzeditor zu verlassen und alle erforderlichen Werte zu einem späteren Zeitpunkt anzugeben. Solange die fehlenden Felder einer Instanz nicht angegeben wurden, ist es nicht möglich, diese Instanz zu aktivieren.

Eine Instanz aktivieren

Sie können jede Instanz separat aktivieren. Eine Instanz kann sich zum Beispiel im Produktionsmodus befinden und eine andere im Testmodus. Ausführliche Informationen zu jedem Modus finden Sie in „Ein Modell aktivieren“ auf Seite 56; diese Informationen gelten sowohl für reguläre Modelle als auch für Vorlagenmodelle.

Wenn Sie eine Instanz aktivieren, werden zuerst alle Änderungen für diese Instanz gespeichert. Erst danach wird die Instanz aktiviert.

Wenn eine Instanz aktiviert wird, dann werden die Werte der Vorlagenparameter, sofern vorhanden, genommen und auf diejenigen Blockparameter im Modell angewendet, an die die Vorlagenparameter gebunden sind. Wenn für einen Vorlagenparameter kein Wert angegeben wurde, dann wird der Standardwert für diesen Vorlagenparameter verwendet, falls es einen gibt. Wenn im Fall eines erforderlichen Vorlagenparameters kein Wert angegeben wurde, dann kann die Instanz nicht aktiviert werden.

Wenn eine Instanz aktiv ist, können Sie die Werte der Vorlagenparameter oder den Modus nicht mehr ändern, ohne die Instanz zuerst zu deaktivieren. Sobald eine Instanz aktiv ist, ist das Modell schreibgeschützt und kann erst wieder geändert werden, nachdem alle Instanzen deaktiviert wurden.

Eine Instanz aktivieren

1. Klicken Sie in der **Modus**-Spalte des Instanzeditors auf das Dropdown-Listenfeld der Instanz, die Sie aktivieren möchten und wählen Sie eine der folgenden Optionen aus: **Produktion**, **Test** oder **Simulation**.

   Sie können keine Instanzen aktivieren, die sich im Entwurfsmodus befinden.

2. Wenn Sie den Simulationsmodus ausgewählt haben, klicken Sie auf das nun angezeigte Kalendersymbol, geben Sie den gewünschten Zeitraum an und klicken Sie auf **Anwenden**. Siehe auch „Simulationsparameter“ auf Seite 145.

3. Wenn die Umschaltfläche in der **Status**-Spalte gerade **Inaktiv** anzeigt, klicken Sie auf diese Umschaltfläche, um den Status in **Aktiv** zu ändern. Für den Simulationsmodus können Sie den Status nur dann auf **Aktiv** setzen, wenn Sie einen gültigen Zeitraum angegeben haben.

   Wenn ein Fehler auftritt, wird rechts in der Tabelle ein Fehlersymbol angezeigt und die Instanz kann nicht aktiviert werden. Klicken Sie auf das Fehlersymbol, um weitere Informationen zu erhalten.

Eine Instanz deaktivieren

Sie können jede Instanz deaktivieren, die sich gerade im Produktions-, Test- oder Simulationsmodus befindet und deren Umschaltfläche in der **Status**-Spalte des Instanzeditors zur Zeit **Aktiv** anzeigt.

Wenn Sie eine Instanz deaktivieren, dann wird die Instanz gestoppt und und eingehende Daten werden nicht mehr verarbeitet. Jeder in der Instanz aufgebaute Zustand geht verloren. Für den Simulationsmodus bedeutet dies, dass die Instanz gestoppt wird, bevor alle historischen Daten aus dem angegebenen Zeitraum durchgespielt wurden.

Eine Instanz deaktivieren

- Klicken Sie in der **Status**-Spalte des Instanzeditors auf die Umschaltfläche der Instanz, die Sie tdeaktivieren wollen, so dass **Inaktiv** auf dieser Umschaltfläche angezeigt wird.

Die Instanzen filtern und sortieren

Wenn Sie eine lange Liste von Instanzen haben, dann können Sie die gewünschte Instanz leicht finden, indem Sie einen Wert im Suchfeld eingeben. Sie können auch nur einen Teil des Werts eingeben. Dies durchsucht alle Eingabefelder im Instanzeditor und listet nur die Instanzen (Zeilen) auf, die diesen Wert enthalten. Alle Werte, die dem Filter entsprechen, werden hervorgehoben. Die Suchkriterien unterscheiden nicht zwischen Groß- und Kleinschreibung. Wenn Suchkriterien gerade angewendet werden, dann wird ein × im Suchfeld angezeigt. Wenn Sie dies anklicken, dann werden die Suchkriterien gelöscht und alle verfügbaren Instanzen werden wieder angezeigt.

Sie können die Spalten der Tabelle auch sortieren. Klicken Sie hierzu auf die Pfeile, die angezeigt werden, wenn Sie mit der Maus über eine Spaltenüberschrift fahren. Die Instanzen werden dann nach den Werten in dieser Spalte sortiert (zum Beispiel alphabetisch oder nach Zahlen). Durch erneutes Klicken wird die Spalte in die entgegengesetzte Richtung sortiert. Die Felder mit erforderlichen Werten in dieser Spalte, die noch nicht angegeben wurden, können daher entweder ganz oben oder ganz unten in der Spalte angezeigt werden. Sie können die Instanzen auch alphabetisch nach Modus und Status sortieren, zum Beispiel, um die aktiven Instanzen ganz oben anzuzeigen.

Durch das Editieren eines Werts wird die Anzeige der Zeilen in der Tabelle nicht beeinflusst. Wenn Sie die Suchkriterien und Sortierung erneut anwenden möchten, müssen Sie die Instanzen speichern und neu laden.

Durch das Erstellen einer neuen Instanz wird die Anzeige der Zeilen in der Tabelle nicht beeinflusst. Wenn Sie nach dem Sortieren eine Zeile für die neue Instanz hinzufügen, dann wird die Zeile immer am Ende der Tabelle hinzugefügt, außer wenn Sie die Instanzen neu laden.

Anmerkung:

Sie können die Vorlagenparameter auch im Dialogfeld "Vorlagenparameter" neu sortieren (siehe „Vorlagenparameter verwalten“ auf Seite 78). Dies wirkt sich auf die Reihenfolge aus, in der sie im Instanzeditor angezeigt werden.

Eine Instanz duplizieren

Sie können jede Instanz (Zeile) duplizieren, die zur Zeit im Instanzeditor aufgelistet wird. Die Originalinstanz und ihr Duplikat haben dann beide die gleichen Vorlagenparameterwerte und den gleichen Modus. Die duplizierte Instanz ist jedoch immer inaktiv, auch wenn die Originalinstanz aktiv ist.

Eine Instanz duplizieren

- Klicken Sie im Instanzeditor auf das Aktionsmenü der Instanz, die Sie duplizieren möchten, und klicken Sie dann auf **Duplizieren**.

  Am unteren Ende der Tabelle wird sofort eine neue Zeile für die duplizierte Instanz angezeigt.

Eine Instanz entfernen

Sie können jede Instanz entfernen, die zur Zeit im Instanzeditor aufgelistet wird. Wenn die Instanz gerade aktiv ist, dann wird sie zuerst deaktiviert und danach entfernt.

Eine Instanz entfernen

1. Klicken Sie im Instanzeditor auf das Aktionsmenü der Instanz, die Sie entfernen möchten, und klicken Sie dann auf **Entfernen**.

2. Klicken Sie im anschließend angezeigten Dialogfeld auf **Entfernen**, um das Entfernen zu bestätigen.

Die Instanzen speichern

Sie können die Instanzen auch dann speichern, wenn es noch Zeilen gibt, in denen erforderliche Informationen noch nicht eingegeben wurden. Dies ist hilfreich, wenn Sie diese Informationen zu einem späteren Zeitpunkt eingeben möchten.

Anmerkung:

Wenn Sie eine Instanz aktivieren, werden alle Ihre letzten Änderungen automatisch gespeichert. Siehe auch „Eine Instanz aktivieren“ auf Seite 94.

Die Instanzen speichern

- Klicken Sie in der Symbolleiste des Instanzeditors auf **Speichern**.

Dieser Befehl ist nur dann aktiviert, wenn Änderungen an den Instanzen vorgenommen wurden. Es werden nur die Instanzen gespeichert, deren Zeilen mit einer Hintergrundfarbe hervorgehoben sind.

Die Instanzen neu laden

Sie können die Anzeige aktualisieren, um den neuesten Zustand aller Instanzen anzuzeigen oder um festzustellen, ob die aktiven Instanzen in einen fehlerhaften Zustand übergegangen sind.

Die Instanzen neu laden

- Klicken Sie in der Symbolleiste des Instanzeditors auf **Neu laden**.

  Wenn es Änderungen gibt, die Sie noch nicht gespeichert haben, dann werden Sie aufgefordert, diese vor dem erneuten Laden zu speichern.

Den Instanzeditor verlassen

Wenn Sie den Instanzeditor mit der entsprechenden Schaltfläche in der Symbolleiste verlassen, kehren Sie zum Modellmanager zurück.

Den Instanzeditor verlassen

- Klicken Sie in der Symbolleiste des Instanzeditors auf ✕.

  Wenn es noch Änderungen gibt, die Sie noch nicht gespeichert haben, dann werden Sie aufgefordert, diese vor dem Verlassen zu speichern.

7 Verbindungen und Blöcke

- Über eine Verbindung gesendete Werte 100
- Typumwandlungen 106
- Verarbeitungsreihenfolge der Verbindungen 108
- Verbindungseinschränkungen 109
- Ein- und Ausgänge der Blöcke 110
- Häufig vorkommende Blockeingänge und Parameter 110
- Eingabeblöcke und Ereignissteuerung 112
- Ausgabeblöcke und Ereignissteuerung 113
- Fragmenteigenschaften auf Verbindungen 114
- Schlüssel zur Identifizierung einer Ereignisreihe 114

Über eine Verbindung gesendete Werte

Die Blöcke in einem Modell werden miteinander verknüpft, in dem Sie Verbindungen von den Blockausgängen zu den Blockeingängen ziehen.

Anmerkung:

Die Ein- und Ausgänge eines Blocks werden auch *Eingangsports* und *Ausgangsports* genannt. Siehe auch „Zwei Blöcke verbinden“ auf Seite 74.

Über die Verbindungen werden Signale und Werte zwischen den Blöcken übertragen. Der übertragene Wert kann von einem der folgenden Typen sein. Der Typ ist abhängig von dem verbundenen Blockausgang.

Typ	Beschreibung
`boolean`	Der Wert ist entweder wahr (true) oder falsch (false). Ein boolescher Wert bleibt solange wahr oder falsch bis er geändert wird.
`float`	Ein numerischer Wert, der eine Bruchzahl enthalten kann (und mit fester Genauigkeit verarbeitet werden kann). Ein Float-Wert behält seinen aktuellen Wert bei, bis er geändert wird.
`string`	Ein textueller Wert. Ein String behält seinen aktuellen Wert bei, bis er geändert wird.
`pulse`	Ein Signal an einem Zeitpunkt. Pulse-Signale sind nur momentan aktiv. Im Gegensatz zu den oben genannten Typen repräsentiert der `pulse`-Typ nur einen einzigen Zeitpunkt. Siehe auch „Der pulse-Typ“ auf Seite 105.
`any`	Ein Wert, der von einem beliebigen oben genannten Typ sein kann. Siehe auch „Der any-Typ“ auf Seite 106.

Der Verbindungstyp ist abhängig von dem Ausgang des Blocks, mit dem die Verbindung verknüpft ist. Ausführliche Informationen hierzu finden Sie in der Blockreferenz. Dort finden Sie auch Informationen über den Typ (oder die unterstützten Typen) eines Blockeingangs.

Werttypen

Die folgenden Typen werden als *Werttypen* bezeichnet:

- `boolean`
- `string`
- `float`
- `any` wenn darin ein Wert vom Typ `boolean`, `string` oder `float` enthalten ist

Werttypen sind bei der Modellierung von Messungen nützlich (zum Beispiel bei Sensorwerten), die periodisch gelesen oder als Stichprobe genommen werden können. Zwischen den Ablesungen

hat die zu messende physikalische Eigenschaft (wie zum Beispiel die Temperatur) immer noch einen gewissen Wert, da es sich um eine kontinuierliche Eigenschaft handelt. Aus praktischen Gründen kann es sein, dass ein Sensor keine kontinuierliche Ausgabe liefert und dass statt dessen nur eine periodische Stichprobe genommen wird. Oder es kann sein, dass nur dann neue Messwerte geliefert werden, wenn sich der zu messende Wert ändert (im Rahmen der vom Sensor bereitgestellten Messauflösung). Die Blöcke verwenden zwischen den Stichproben immer den neuesten empfangenen Wert. Sie gehen davon aus, dass ein Wert so bleibt, wie er zuletzt gelesen wurde, und zwar solange bis ein neuer Wert empfangen wird.

Stellen Sie sich zum Beispiel zwei Temperatursensoren vor. Der eine Sensor liefert alle 10 Sekunden einen neuen Messwert. Der andere Sensor liefert nur dann einen neuen Wert, wenn sich die Temperatur um 0,5 Grad verändert. Wenn wir diese Sensoren mit einem **Differenz**-Block verbinden, dann könnten wir Eingabewerte wie in der folgenden Tabelle erhalten, mit einem entsprechenden Ergebnis des **Absolute Differenz**-Ausgangsports des **Differenz**-Blocks:

Zeit	Sensor 1 (liest alle 10 Sekunden)	Sensor 2 (Ausgabe bei Änderung von 0,5 Grad)	Differenz-Block: Absolute Differenz
10:00:00	20,0		
10:00:03		22,0	2
10:00:10	20,0		2
10:00:20	20,0		2
10:00:23		22,5	2,5
10:00:28		23,0	3
10:00:30	21,1		1,9
10:00:35		23,5	2,4
10:00:40	22,8	24,0	1,2

Beachten Sie, dass zwei Eingaben für denselben Block (zu unterschiedlichen Eingangsports des Blocks) mit demselben Zeitstempel nur eine einzige Ausgabe erzeugen. Für jede Verbindung innerhalb eines Modells (und für jeden Eingabeblock) kann es zu einem bestimmten Zeitpunkt immer nur einen einzigen Wert geben. Ein Eingabeblock kann nicht mehr als eine Ausgabe für denselben Zeitstempel erzeugen. Wenn ein Block mehrere Ereignisse zum selben Zeitpunkt empfängt, dann ist es unbestimmt, welches der Ereignisse genommen wird.

Im Allgemeinen wird einer Verbindung keine Bedeutung zugemessen, wenn sie dieselben `boolean`-, `float`- oder `string`- Werte wie zuvor empfängt. Die meisten Blöcke ändern das Verhalten in diesem Fall nicht. Dies gilt für alle arithmetischen Blöcke, wie zum Beispiel auch für den **Differenz**-Block im obigen Beispiel: bei wiederholten Ablesungen von Sensor 1 ist die Ausgabe immer noch 2. Es gibt jedoch einige Ausnahmen, wie zum Beispiel den **Fehlende Daten**-Block, wenn das Kontrollkästchen **Wiederholte Eingaben ignorieren** nicht markiert wurde (`false`).

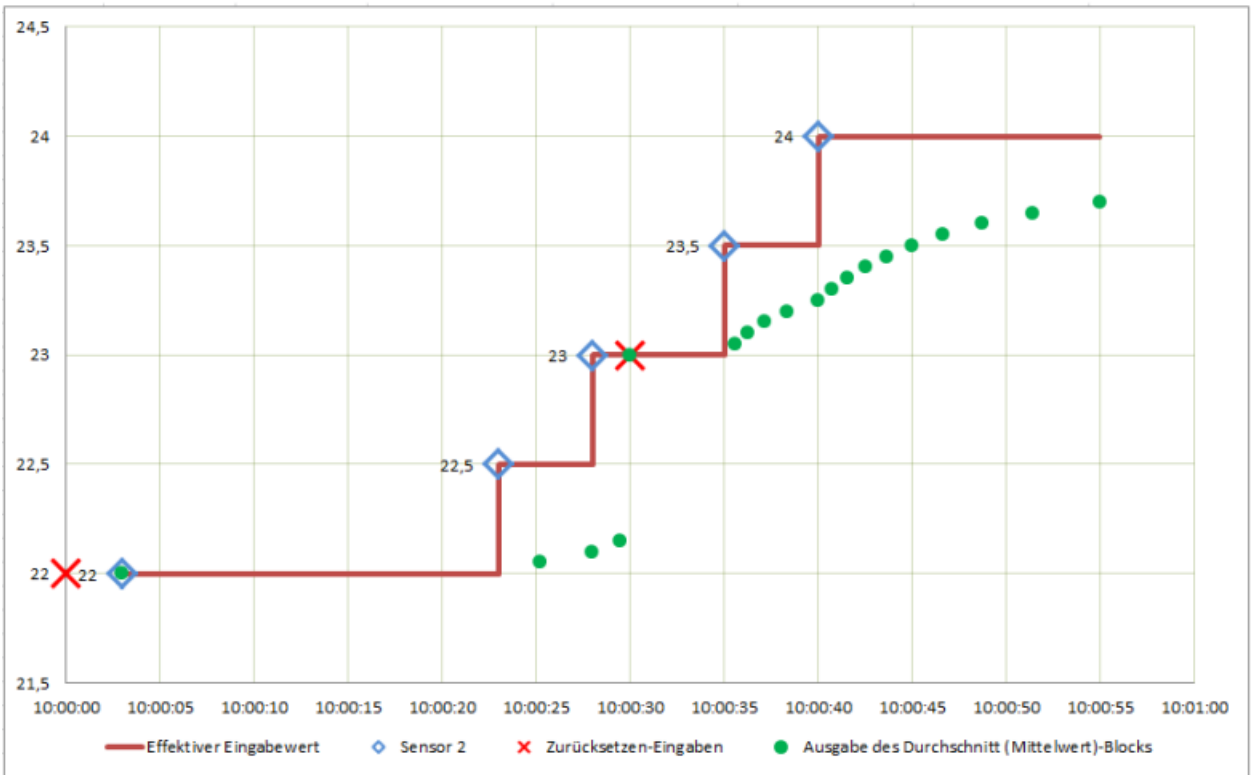
Wenn ein einzelner Block einen numerischen Werteingang und `pulse`-Signale (wie zum Beispiel zum Zurücksetzen) hat, dann bedeutet das Fehlen eines neuen Werts bei einem erneuten

`pulse`-Signal, dass der Wert immer noch so behandelt wird als hätte er denselben Wert. Wenn also ein **Durchschnitt (Mittelwert)**-Block zurückgesetzt wird, dann ist dessen Ausgabe mit der zuletzt empfangenen Eingabe identisch (unter der Voraussetzung, dass er seit dem Starten des Modells eine Eingabe empfangen hat). Für das nachstehenden Beispiel wurde die Dauer des **Durchschnitt (Mittelwert)**-Blocks nicht definiert und der Schwellenwert für die Ausgabe wurde auf 0,05 eingestellt. Dies bedeutet, dass der Block auch dann eine neue Ausgabe generiert, wenn er keine neue Eingabe erhält (siehe „Häufig vorkommende Blockeingänge und Parameter“ auf Seite 110).

Zeit	Zurücksetzen-Signal	Sensor 2	Ausgabe des Durchschnitt (Mittelwert)-Blocks	Anmerkungen
10:00:00	Zurücksetzen			Keine Ausgabe. Es ist noch kein Eingabewert vorhanden.
10:00:03		22,0	22,00	Ohne Historie entspricht der Ausgabewert dem Eingabewert.
10:00:23		22,5		Bis zu diesem Zeitpunkt waren alle Werte 22. Der Durchschnittswert ist immer noch 22 (somit wird keine neue Ausgabe generiert).
10:00:25.22			22,05	Ein Durchschnitt von 20 Sekunden bei einem Wert von 22, und 2,22 Sekunden bei einem Wert von 22,5.
10:00:28		23,0	22,10	Ein Durchschnitt von 20 Sekunden bei einem Wert von 22, und 5 Sekunden bei einem Wert von 22,5.
10:00:30	Zurücksetzen		23,00	Die Eingabe ist immer noch 23 (wir haben kein neues Ereignis erhalten). Das Zurücksetzen löscht nur die Historie. Ohne Historie entspricht der Ausgabewert dem Eingabewert.
10:00:35		23,5		
10:00:35.56			23,05	Durchschnitt zu verschiedenen Zeitpunkten, wenn sich die Ausgabe um 0,05 ändert.
10:00:36.25			23,10	
10:00:37.14			23,15	
10:00:38.33			23,20	
10:00:40		24,0	23,25	Durchschnitt von 5 Sekunden bei einem Wert von 23 (vom Zurücksetzen um :30 bis :35), und 5

Zeit	Zurücksetzen-Signal	Sensor 2	Ausgabe des Durchschnitt (Mittelwert)-Blocks	Anmerkungen
				Sekunden bei einem Wert von 23,5 (von :35 bis :40).

Die folgende Grafik zeigt die Eingaben für den **Durchschnitt (Mittelwert)**-Block und die Ausgabe dieses Blocks:



Beachten Sie wie der effektive Eingabewert solange unverändert bleibt, bis eine neue Messeingabe erfolgt und wie der **Durchschnitt (Mittelwert)**-Block auf diesen effektiven Wert wirkt (die rote Linie oben in der Grafik). Beim Zurücksetzen gibt der Block den aktuellen effektiven Eingabewert aus, der beim zweiten Zurücksetzen um 10:00:30 den Wert 23 hat. Wenn der Parameter **Ausgabe-Schwellenwert** definiert wurde, dann können neue Ausgaben sogar dann generiert werden, wenn keine neue Eingabe vorliegt; der Ausgabewert nähert sich dann asymptotisch dem letzten Eingabewert. Denken Sie daran, dass sich dieses Verhalten von Apama-Abfragen oder Stream-Abfragen unterscheidet.

Hätte man den **Durchschnitt (Mittelwert)**-Block mit einem Fenster von 10 Sekunden konfiguriert, dann würde sich das Fenster folgendermaßen auswirken:

Zeit	Zurück-setzen-Signal	Sensor 2	Effektiver Eingabe-wert	Ausgabe des Durchschnitt (Mittelwert)-Blocks	Werte in der Historie des Fensters	Anmerkungen
10:00:00	Zurück-setzen					
10:00:03		22	22	22,00		Erster Wert nach dem Start: Das Fenster ist leer. Daher wird der Eingabewert vom **Durchschnitt (Mittelwert)**-Block für die Ausgabe benutzt.
10:00:23		22,5	22,5		22	
10:00:23 - 10:00:28			22,5	Steigt von 22,00 auf 22,20	22, 22,5	Der Anteil des Fensters, der bei 22 oder 22,5 ist, ändert sich mit der Zeit. Daher ändert sich die Ausgabe.
10:00:28		23	23	22,25	22, 22,5	
10:00:28 - 10:00:30			23	Steigt von 22,25 auf 22,40	22, 22,5, 23	
10:00:30	Zurück-setzen		23	23,00		Das Fenster wird zurückgesetzt und ist jetzt somit leer. Der aktuelle (effektive) Eingabewert ist 23. Daher wird dieser Wert vom **Durchschnitt (Mittelwert)**-Block für die Ausgabe benutzt.
10:00:35		23,5	23,5		23	
10:00:35 - 10:00:40			23,5	Steigt von 23,00 auf 23,20	23, 23,5	
10:00:40		24	24	23,25	23, 23,5	Da Fenster ist jetzt voll (10 Sekunden seit dem Zurücksetzen).

Zeit	Zurück-setzen-Signal	Sensor 2	Effektiver Eingabe-wert	Ausgabe des Durchschnitt (Mittelwert)-Blocks	Werte in der Historie des Fensters	Anmerkungen
10:00:40 - 10:00:45			24	Steigt von 23,25 auf 23,75	23, 23,5, 24	
10:00:45			24	23,75	23,5, 24	Der Wert 23 ist nun endgültig im Fenster abgelaufen (dies war der effektive Eingabewert bis 10:00:35, also bis vor 10 Sekunden).
10:00:45 - 10:00:50			24	Steigt von 23,75 auf 24	23,5, 24	
10:00:50			24	24	24	Der Wert 23,5 ist nun endgültig im Fenster abgelaufen (dies war der effektive Eingabewert bis 10:00:40, also bis vor 10 Sekunden). Das Fenster enthält jetzt Messungen aus 10 Sekunden, alle mit dem Wert 24.

In dem Beispiel oben können Sie erkennen, dass der aktuelle Wert erst dann eine Gewichtung im Fenster hat (das heißt, zum Ausgabewert beiträgt) nachdem der Messwert empfangen wurde. Zu dem Zeitpunkt, zu dem der Messwert empfangen wird, hat er im Vergleich zur vorherigen Historie eine Gewichtung von Null. Wie zuvor bleibt der Sensorwert so lange der effektive Eingabewert, bis er durch einen neueren Wert ersetzt wird (beachten Sie, dass dies anders ist als bei Aggregaten mit zeitgesteuerten Fenstern in Apama-Queries oder Stream-Queries). Zum Beispiel: der Block hat von 10:00:35 bis 10:00:40 einen effektiven Eingabewert von 23,5, und der Wert 23,5 ist erst um 10:00:50 endgültig aus dem Fenster abgelaufen, also 10 Sekunden, nachdem er aufgehört hat, der aktuelle effektive Eingabewert zu sein (und nicht 10 Sekunden, nachdem er das erste Mal in das Fenster eingetreten ist). Beachten Sie abschließend Folgendes: wenn das Fenster leer ist, dann wird der effektive Eingabewert stattdessen als Ausgabe verwendet, da das Fenster die Länge Null hat.

Der pulse-Typ

Im Gegensatz zu den Werttypen stellt der `pulse`-Typ einen einzigen Zeitpunkt (ein Signal) dar. Dies kann zum Beispiel das Resultat von Folgendem sein:

- ein Benutzer drückt eine Taste für eine momentane Aktion,

- ein Gerät geht in einen anderen Zustand über,
- ein Sensor erkennt eine Person, die durch eine Tür geht,
- ein Heartbeat-Ereignis zeigt an, dass ein Remote-Gerät noch am Leben ist, oder
- ein Block in einem Modell geht in einen anderen Zustand über.

Normalerweise reagiert ein Block auf jedes Pulse-Signal, das an einen der Blockeingänge gesendet wird. Pulse-Signale werden im Allgemeinen dazu verwendet, die Ausgabe eines Modell über einen Ausgabeblock auszulösen oder den Status der Blöcke im Modell zurückzusetzen.

Pulse-Signale sind nur momentan aktiv. In gewisser Hinsicht ähnelt ein solches Signal einem booleschen Wert, der automatisch auf `false` zurückgesetzt wird, nachdem das Modell einen Wert verarbeitet hat.

Wiederholte Pulse-Signale sind in der Regel von Bedeutung, auch wenn sie - je nach Verwendung - nicht unbedingt zu einer Änderung führen. Ein wiederholtes Zurücksetzen eines **Durchschnitt (Mittelwert)**-Blocks bei einem unveränderten Eingabewert führt zum Beispiel dazu, dass der Ausgabewert gleich bleibt.

Der any-Typ

Der `any`-Typ wird bei Blöcken verwendet, die einen Wert eines beliebigen Typs übergeben (zum Beispiel, bei einem **Zeitverzögerung**- oder **Gate**-Block).

Die Werte des `any`-Typs können einen Werttyp oder einen `pulse`-Typ darstellen.

Typumwandlungen

Es ist zulässig, einen Blockausgang mit einem Blockeingang zu verbinden, wenn beide vom selben Typ sind. Die meisten anderen Verbindungen sind ebenfalls zulässig; diese führen zu den nachstehend beschriebenen Typumwandlungen. Ein bedeutet, dass die Verbindung nicht zulässig ist; wenn Sie versuchen ein Modell mit einer solchen Verbindung zu aktivieren, dann wird dies fehlschlagen.

		Einen Block mit einem Ausgangsport vom Typ				
		pulse	**boolean**	**float**	**string**	**any**
mit einem Eingangs-port von diesem Typ verbinden	**pulse**		Ein Pulse-Signal erfolgt, wenn sich die Ausgabe in wahr ändert	Ein Pulse-Signal erfolgt, wenn sich die Ausgabe ändert	Ein Pulse-Signal erfolgt, wenn sich die Ausgabe ändert	Ein Pulse-Signal erfolgt, wenn sich die Ausgabe ändert (außer wenn sie sich in falsch ändert)

		Einen Block mit einem Ausgangsport vom Typ				
		pulse	**boolean**	**float**	**string**	**any**
	boolean	Wahr wenn ein Pulse-Signal erfolgt, andernfalls falsch	✓	Wahr wenn dies nicht Null ist	Wahr wenn dies keine leere Zeichen-folge ist	Wahr wenn der Wert nicht Null/leer ist
	float	✗	0 für falsch, 1 für wahr	✓	✗	Zulässig, wenn der Wert vom Typ float oder boolean ist, andere Werte schlagen zur Laufzeit fehl
	string	✗	"true" oder "false"	Die Zahl wird in eine Zeichen-folge um-gewandelt (kann in wissen-schaftlicher Notation sein)	✓	Zeichen-folge (kann in wissen-schaftlicher Notation sein)
	any	✓	✓	✓	✓	✓

Es sind nur Umwandlungen zulässig, die immer zu einem Erfolg führen. Zeichenfolgen werden nicht in Float-Werte konvertiert; aber auch wenn die Eingabeumwandlung in manchen Fällen funktioniert, kann nicht garantiert werden, dass dies immer funktioniert.

In vielen Fällen müssen Sie sich über die Typumwandlung oder wo eine Verbindung sinnvoll wäre keine Gedanken machen. Jede erforderliche Typumwandlung erfolgt automatisch.

Einige Blöcke akzeptieren unterschiedliche Eingabetypen und können - in Abhängigkeit von den Eingabetypen - ihren Ausgabetyp oder ihr Verhalten ändern. So kann der logische **ODER**-Block zum Beispiel entweder mit booleschen Eingaben oder `pulse`-Signalen arbeiten, und der Ausgabetyp ist dann derselbe wie die Eingabetypen.

In einigen Fällen ist es wünschenswert, einen Wert als einen bestimmten Typ interpretieren zu lassen. In diesem Fall können Sie die Umwandlung in einen bestimmten Typ mit einem speziellen Block erzwingen. So kann ein **Pulse**-Block zum Beispiel boolesche Werte oder Float-Werte in Pulse-Signale umwandeln (siehe die Tabelle oben). Das bedeutet, dass für `boolean` ein `pulse` erzeugt

wird, wenn sich der boolesche Wert in wahr ändert, und dass für `float` ein `pulse` erzeugt wird, wenn sich der Wert ändert. Wenn Sie also zwei Float-Ausgänge direkt mit einem **ODER**-Block verbinden, dann wird immer dann eine wahre boolesche Ausgabe erzeugt, wenn eine der Float-Ausgaben nicht Null ist. Wenn Sie alternativ zwei Float-Ausgänge mit einem **Pulse**-Block verbinden und diese dann mit den Eingängen eines **ODER**-Blocks verbinden, dann wird immer dann ein Pulse-Signal gesendet, wenn einer der Float-Ausgänge den Wert ändert. Dies ist das Standardverhalten des **Pulse**-Blocks.

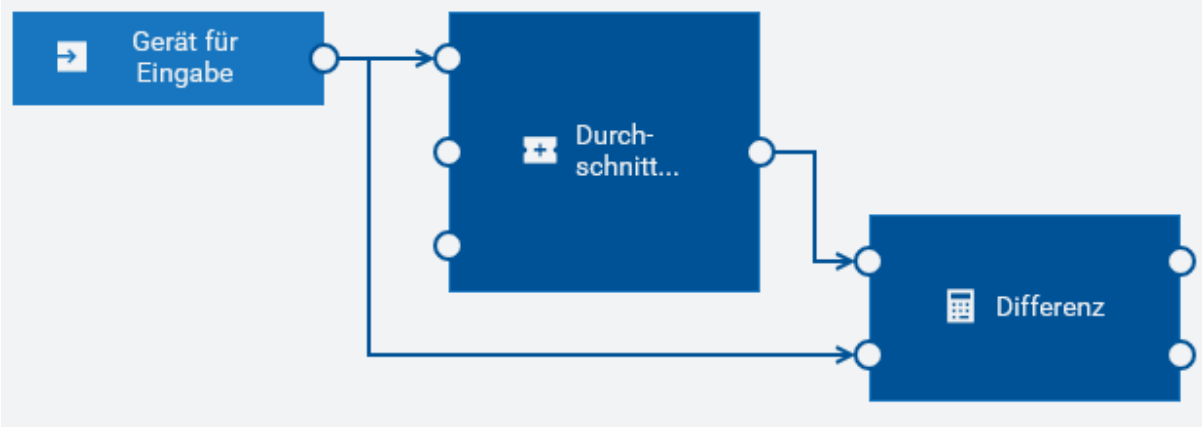
Mit dem **Pulse**-Block sind verschiedene Arten von Pulse-Signal-Umwandlungen möglich. Dies ist abhängig von der Einstellung seines **Modus**-Parameters. Diese Umwandlungen sind in der folgenden Tabelle beschrieben:

		Einen Block mit einem Ausgangsport vom Typ				
		pulse	**boolean**	**float**	**string**	**any**
mit einem Pulse-Block in diesem Modus verbinden	**Bei Wertänderung (Standard)**	✓	Ein Pulse-Signal erfolgt, wenn sich die Ausgabe in wahr ändert	Ein Pulse-Signal erfolgt, wenn sich die Ausgabe ändert	Ein Pulse-Signal erfolgt, wenn sich die Ausgabe ändert	Ein Pulse-Signal erfolgt, wenn sich die Ausgabe ändert (außer wenn sie sich in falsch ändert)
	Bei jeder Eingabe	✓	Ein Pulse-Signal erfolgt bei jeder Eingabe	Ein Pulse-Signal erfolgt bei jeder Eingabe	Ein Pulse-Signal erfolgt bei jeder Eingabe	Ein Pulse-Signal erfolgt bei jeder Eingabe
	Bei Nicht-Null-Werten	✓	Ein Pulse-Signal erfolgt bei jeder wahren Eingabe	Ein Pulse-Signal erfolgt bei jeder Eingabe, die nicht Null ist	Ein Pulse-Signal erfolgt bei jeder Eingabe, die nicht leer ist	Ein Pulse-Signal erfolgt in Abhängigkeit vom Typ des Werts, wie in den Zellen links beschrieben

Verarbeitungsreihenfolge der Verbindungen

Wenn an einem Block mehrere Eingänge angeschlossen sind, dann werden zuerst alle Eingaben berechnet, und erst danach führt der Block die Berechnungen mit diesen Eingaben aus. Es kann passieren, dass die Eingaben für einen Block nicht direkt aufeinander folgen (wie in dem Beispiel mit den zwei Temperatursensoren in „Werttypen“ auf Seite 100). In diesem Fall verwendet der Block den neuesten Wert als Eingabe für die Werttypen.

Wenn ein einzelner Wert auf zwei oder mehr Verbindungen gesendet wird, die beide zum selben Block führen, dann führt der Block die Berechnungen basierend auf dem letzten Wert für beide Verbindungen aus. Dies gewährleistet ein konsistentes Verhalten, wenn mehrere Verbindungen zu einem einzelnen Block vorhanden sind. Beispiel:



Wenn der Wert von dem Gerät empfangen wird, dann führt der **Durchschnitt (Mittelwert)**-Block die Berechnung durch bevor der **Differenz**-Block die Differenz zwischen dem Wert und seinem Durchschnitt berechnet.

Verbindungseinschränkungen

Ein Ausgang eines Blocks kann mit mehreren anderen Blöcken verbunden werden. Ein Eingang eines Blocks kann aber nur eine einzige Verbindung haben.

Sie können die Ein- oder Ausgänge eines Blocks auch unverbunden lassen, wenn dies nicht erforderlich ist (im **Durchschnitt (Mittelwert)**-Block in dem Beispiel in „Verarbeitungsreihenfolge der Verbindungen“ auf Seite 108 sind die **Probe**- und **Zurücksetzen**-Ports auch nicht verbunden).

Es ist nicht möglich, Kreisläufe mit den Verbindungen zu erzeugen. Das heißt, dass der Ausgang eines Blocks nicht mit Folgendem verbunden werden kann:

- mit einem Eingang desselben Blocks, oder
- mit dem Eingang eines Blocks, der direkt oder indirekt mit einem der Eingänge des Quellblocks verbunden ist.

Nehmen wir zum Beispiel an, dass wir drei Blöcke haben: Block1, Block2 und Block3. Das Modell würde in den folgenden Fällen einen Kreislauf enthalten:

- Der Ausgang von Block1 ist mit dem Eingang von Block2 verbunden, und der Ausgang von Block2 ist mit dem Eingang von Block1 verbunden.
- Der Ausgang von Block1 ist mit dem Eingang von Block2 verbunden, der Ausgang von Block2 ist mit dem Eingang von Block3 verbunden, und der Ausgang von Block3 ist wieder mit dem Eingang von Block1 verbunden.

Es gibt viele Verbindungsmöglichkeiten, die zu Kreisläufen in einem Modell führen könnten. Der Modelleditor hindert Sie jedoch am Erstellen von Kreisläufen.

Ein- und Ausgänge der Blöcke

Viele Blöcke haben Ein- oder Ausgänge, die nicht verwendet werden müssen.

Einige Blöcke erzeugen mehrere unterschiedliche Ausgaben, und ein Modell benötigt möglicherweise nur einige dieser Ausgaben.

Einige Blöcke haben Eingänge, die nicht verwendet werden müssen. Das sind insbesondere Eingänge des `pulse`-Typs. Es ist in Ordnung diese Eingänge nicht zu verbinden. Die mit diesen Eingängen (zum Beispiel **Zurücksetzen**, siehe „Häufig vorkommende Blockeingänge und Parameter“ auf Seite 110) verbundenen Operationen werden dann niemals ausgelöst.

Ein Block kann bei Bedarf erkennen, welche Eingänge verbunden sind. Der **UND**-Block hat zum Beispiel fünf Eingänge, aber um die Ausgabe `true` zu erzeugen müssen nur die verbundenen Eingänge `true` sein.

Häufig vorkommende Blockeingänge und Parameter

In der folgenden Liste finden Sie die Namen von häufig vorkommenden Eingangsports. Diese Ports werden auf der linken Seite eines Blocks angezeigt.

- **Wert**-Port

  Die meisten Berechnungsblöcke haben einen Haupteingangsport mit dem Namen **Wert**. Dies ist der Wert, auf dem der Block die Hauptberechnung durchführt.

- **Wert 1**- und **Wert 2**-Ports

  Die Blöcke können eine Reihe von ähnlichen Eingängen mit den Namen **Wert 1**, **Wert 2** und so weiter haben. Solche Eingänge finden Sie beim **Differenz**-Block (siehe auch das Beispiel unter „Werttypen“ auf Seite 100) oder bei den logischen **UND**- und **ODER**-Blöcken. Normalerweise ist es egal, welcher dieser Eingänge verwendet wird.

- **Zurücksetzen**-Port

  Die Blöcke, die einen internen Zustand verwalten, können auch einen **Zurücksetzen**-Eingang haben, der normalerweise vom Typ `pulse` ist. Dieser Eingang muss nicht verbunden werden. Sie können ihn aber benutzen, um explizit zu steuern, für welchen Wertebereich ein Block eine Berechnung durchführen soll. Zum Beispiel kann ein Modell, das die Fahrten eines Fahrzeugs überwacht, beim Starten des Motors zurückgesetzt werden, weil dies auf den Beginn einer Fahrt hinweist. Siehe auch das Beispiel in „Werttypen“ auf Seite 100, bei dem die Benutzung des **Zurücksetzen**-Eingangsports veranschaulicht wird.

- **Probe**-Port und der Parameter **Ausgabe-Schwellenwert**

  Die Blöcke berechnen ihre Ausgabe normalerweise neu, wenn eine neue Eingabe empfangen wird. Einige Blöcke können die Ausgabe auch erst eine bestimmte Zeit nach dem Datenempfang generieren, entweder aufgrund von Zeitverzögerungsparametern (zum Beispiel mit den **Fehlende Daten**- oder **Zeitverzögerung**-Blöcken) oder weil sich die Ausgabe mit der Zeit ändert obwohl der Eingabewert konstant bleibt. So generiert zum Beispiel der **Integral**-Block mit einer positivem Eingabe solange eine ständig steigende Ausgabe bis das Fenster voll ist

(oder eine unendliche Ausgabe, wenn keine Dauer eingestellt wurde und der Block das Integral über ein unbegrenztes Fenster berechnet).

Wie bei realen Sensoren ist es nicht zweckmäßig, eine sich ständig ändernde Ausgabe zu generieren. Zusätzlich zum Generieren einer Ausgabe bei einem geänderten Eingabewert können manche Blöcke auch einen **Probe**-Eingangsport haben, der den Block zu einer Neuauswertung veranlasst und eine neue Ausgabe auch dann generiert, wenn der Eingangsport keinen neuen Wert erhalten hat und die Ausgabe sich nicht erheblich geändert hat. Dies ist nützlich, wenn es einen bestimmten Zeitpunkt gibt, zu dem die Ausgabe des Blocks berechnet werden soll, weil die Ausgabe zu einem späteren Zeitpunkt im Modell verwendet werden soll.

Alternativ können solche Blöcke einen **Ausgabe-Schwellenwert**-Parameter haben, mit dem gesteuert wird, wie oft die Ausgabe neu berechnet werden soll. Wenn dieser Parameter gesetzt wurde, dann bestimmt der Block an Hand des Ausgabeschwellenwerts, wann sich die Ausgabe ändern soll. Wenn dieser Fall eintritt, auch wenn dies nicht das Resultat eines neuen Eingabewerts ist, dann generiert der Block einen Ausgabewert.

Der **Ausgabe-Schwellenwert**-Parameter sollte unter Berücksichtigung der möglichen Fehlergrenzen des Eingabewerts festgelegt werden (in der realen Welt haben physikalische Sensoren eine begrenzte Genauigkeit und Fehlerfreiheit bei den von ihnen gemessenen Eigenschaften) und welche Genauigkeit bei der Ausgabe erforderlich ist.

Vermeiden Sie zu große oder zu kleine Werte im **Ausgabe-Schwellenwert**-Parameter. Wenn die Werte zu groß sind, dann generiert der Block keine neue erforderliche Ausgabe (außer wenn der **Probe**-Port benutzt wird). Wenn die Werte zu klein sind, dann beschränkt der Block die Häufigkeit der Ausgabe. Sie können dies ändern, indem Sie eine `POST`-Anfrage an Cumulocity IoT senden, die den Wert des Keys `minimum_wait_time_secs` ändert. Ausführliche Informationen finden Sie unter „Konfiguration“ auf Seite 157.

Die Skala der geeigneten Werte variiert in Abhängigkeit von der Größe des Eingabewerts. Wenn der **Ausgabe-Schwellenwert**-Parameter nicht gesetzt ist, dann generiert der Block nur dann neue Ausgaben, wenn er eine Eingabe empfängt (dies kann sinnvoll sein, wenn er häufige Eingaben für einen Wert erhält oder wenn der **Probe**-Eingangsport verwendet wird).

- Parameter **Zeitstempel ignorieren**

  Bei Messwerten, Ereignissen und Alarmen von Cumulocity IoT verwendet der Block standardmäßig den Quellzeitstempel der Eingabe. Der Block ordnet die Eingabe basierend auf dem Zeitstempel neu an (siehe auch „Eingabeblöcke und Ereignissteuerung“ auf Seite 112), er löscht jedoch die Ereignisse, bei denen es zu einer zu großen Verzögerung kommt. Wenn dieses Verhalten nicht erwünscht ist (zum Beispiel, wenn die Uhr eines Geräts schlecht synchronisiert ist oder wenn die Daten von einem Gerät möglicherweise verzögert ankommen), können Sie dieses Verhalten deaktivieren, indem Sie den Parameter **Zeitstempel ignorieren** auswählen. Wenn dieser Parameter ausgewählt ist, dann wird der Zeitstempel der Daten ignoriert, und das Modell verarbeitet die Eingabedaten sobald sie empfangen werden, unabhängig vom Zeitstempel. Dies kann zu anderen Ergebnissen führen als das Standardverhalten bei der Verwendung von Zeitstempeln. Welches Verhalten wünschenswert ist, hängt von der Art des Geräts und seiner Anbindung an Cumulocity IoT ab.

Wenn ein Modell im Simulationsmodus ausgeführt wird, dann wird die Einstellung des Parameters **Zeitstempel ignorieren** ignoriert. Der Block verwendet immer den Quellzeitstempel, so dass bei der Wiedergabe von Simulationsereignissen die Daten garantiert in der richtigen Reihenfolge verarbeitet werden und somit realistische Ergebnisse erzielt werden (es gibt keine Aufzeichnung darüber, wann die Daten empfangen wurden, nur den Quellzeitstempel). Siehe auch „Allgemeine Informationen zum Simulationsmodus“ auf Seite 144.

Eingabeblöcke und Ereignissteuerung

Eingabeblöcke stellen dem Modell Daten aus externen Quellen zur Verfügung (zum Beispiel Messwerte von Cumulocity IoT). Viele Datenquellen haben auf jedem Dateneintrag einen Zeitstempel, der den Zeitpunkt angibt, zu dem eine Messung oder ein Ereignis tatsächlich erfolgt ist. Bei der Übertragung der zu verarbeitenden Daten an das Apama-System kann es zu Verzögerungen kommen, was dazu führt, dass Apama die Ereignisse nicht in der richtigen Reihenfolge empfängt.

Datenquellen mit Zeitstempeln (wie zum Beispiel Messwerte) können neu angeordnet werden. Operationen haben zum Beispiel keine Zeitstempel und werden daher so verarbeitet wie sie empfangen werden, ohne Neuanordnung.

Analytics Builder stellt mehrere Eingabeblocktypen zur Verfügung, die Datenquellen mit Zeitstempeln verwenden. Diese Blocktypen haben den Parameter **Zeitstempel ignorieren**, mit dem Sie die Neuanordnung der Daten deaktivieren können. Die Eingaben werden dann so verarbeitet, wie sie empfangen werden. Siehe auch „Häufig vorkommende Blockeingänge und Parameter“ auf Seite 110.

Die folgende Tabelle gibt an, ob die zur Verfügung stehenden Eingabeblocktypen in der Lage sind, die Daten neu anzuordnen:

Eingabeblocktyp	Neuanordnung ist möglich
Messwerteingabe	Ja
Ereigniseingabe	Ja
Alarmeingabe	Ja
Operationseingabe	Nein
Managed-Object-Eingabe	Nein
Positionseingabe	Ja

Anmerkung:

Der **Positionseingabe**-Block ist ein spezialisierter **Ereigniseingabe**-Block. Sie können auch den **Cron Timer**-Block verwenden, um ein Modell periodisch zu aktivieren. Im Gegensatz zu den oben genannten Blöcken ist der **Cron Timer**-Block nicht mit einem Gerät verknüpft und befindet sich in der **Werkzeug**-Kategorie der Palette.

Bei Datenquellen, die Zeitstempel auf den Dateneinträgen haben, kann der Eingabeblock mit Ereignissen umgehen, die nicht in der richtigen Reihenfolge empfangen wurden. Zu diesem Zweck halten die Eingangsblöcke alle empfangenen Ereignisse in einem Umordnungspuffer und verzögern deren Verarbeitung bis die vordefinierte Verzögerungszeit ab dem Zeitstempel der Quelle abgelaufen ist. Durch das Verzögern der Ereignisverarbeitung relativ zum Zeitstempel der Quelle kann der Eingabeblock die Ereignisse neu anordnen. Der entscheidende Parameter für diesen Prozess ist die Dauer der Zeit, um die die Ereignisse verzögert werden. Sie können die Zeit in Sekunden festlegen, um die die Eingabewerte von den Eingangsblöcken verzögert werden. Senden Sie hierfür eine `POST`-Anfrage an Cumulocity IoT, die den Wert des Keys `timedelay_secs` ändert. Ausführliche Informationen finden Sie unter „Konfiguration“ auf Seite 157.

Die Eingabeblöcke gehen davon aus, dass Ereignisse zwar in einer anderen Reihenfolge geliefert werden können, sie jedoch von Apama gemäß der definierten Verzögerungszeit empfangen werden. Wenn ein Ereignis nach einer Verzögerung von mehr als den definierten Sekunden empfangen wird (das heißt, wenn es eine Differenz gibt zwischen dem Zeitstempel des Ereignisses und der Zeit auf dem System, auf dem Apama ausgeführt wird), dann wird es verworfen, wenn ein Ereignis für denselben Zeitstempel oder einen neueren Zeitstempel bereits vom Modell verarbeitet wurde. Daher ist es möglich, dass ein altes Ereignis von einem Modell verarbeitet, aber von einem anderen Modell verworfen wird.

Wenn der Wert für die Verzögerungszeit zu niedrig eingestellt ist, dann kann eine kleine Verzögerung dazu führen, dass Apama ein Ereignis verwirft, was zu fehlerhaften Ergebnissen führen kann. Je höher der Wert für die Verzögerungszeit ist, desto größer ist die Verzögerung vor der Verarbeitung eines Ereignisses. Daher ist es wichtig, einen geeigneten Wert für die Verzögerungszeit zu finden, der zu der Umgebung passt, in der die Ereignisse an Apama geliefert werden.

Anmerkung:

Ein Ereignis, das über die festgelegte Anzahl von Sekunden hinaus verzögert ist, wird immer dann verworfen, wenn benutzerdefinierte Eingabe- oder Ausgabeblöcke, die mit der API-Version 1 des Analytics Builder Block SDK geschrieben wurden, als Extensionen hinzugefügt werden. Informationen darüber, wie Sie solche Blöcke auf die API-Version 2 migrieren können finden Sie in der Dokumentation unter https://github.com/SoftwareAG/apama-analytics-builder-block-sdk.

Der Korrelator schreibt die Anzahl der verworfenen Ereignisse regelmäßig in das Korrelatorlog. Siehe „Konfiguration“ auf Seite 157 (den Key zur Drosselung der Protokollierung) und „Korrelatorlog anzeigen“ auf Seite 161.

Ausgabeblöcke und Ereignissteuerung

Ausgabeblöcke stellen externen Quellen (wie zum Beispiel Cumulocity IoT) Daten aus dem Modell zur Verfügung (zum Beispiel Messwerte oder Operationen). Ausgabeblöcke können entweder synchrone oder asynchrone Werte erzeugen.

Die Werte eines Ausgabeblocks, der eine synchrone Ausgabe (wie zum Beispiel Messwerte) generiert, können auch von einem anderen Modell zeitsynchron verwendet werden und können von dem Modell zusammen mit anderen Daten aus demselben Zeitstempel verarbeitet werden. Siehe auch „Verbindungen zwischen Modellen“ auf Seite 138.

Die Werte aus einem Ausgabeblock, der eine asynchrone Ausgabe generiert, können auch von einem anderen Modell verwendet werden. Dies kann jedoch nur zeitasynchron geschehen, wenn Daten vom externen System zurückempfangen werden.

Die folgende Tabelle gibt an, ob die Ausgabe der zur Verfügung stehenden Ausgabeblocktypen synchron oder asynchron ist:

Ausgabeblocktyp	**Art der Ausgabe**
Messwertausgabe	Synchron
Ereignisausgabe	Synchron
Alarmausgabe	Synchron
Operationsausgabe	Asynchron
Managed-Object-Ausgabe	Asynchron

Fragmenteigenschaften auf Verbindungen

Jede Verbindung hat einen primären Wert, der vom selben Typ wie die Verbindung ist: `float`, `boolean`, `string` oder `pulse`.

Zusätzlich können einige Blöcke neben dem Wert auch weitere *Fragmente* von Informationen bereitstellen. Dies sind benannte Werteigenschaften. Dies können andere von einem Eingabeblock bereitgestellte Informationen sein, wie zum Beispiel die Einheit, in der eine Messung erfolgt oder zusätzliche Kontextinformationen für eine Datenquelle.

Die meisten Blöcke verarbeiten nur den Hauptwert ihrer Eingangsverbindungen, aber einige Blöcke können auch die Werte der Fragmenteigenschaften verwenden und sie an separate Ausgangsports extrahieren (ein Beispiel hierfür ist der **Eigenschaft extrahieren**-Block). Dies gibt mehr Flexibilität bei der Verarbeitung komplexer Daten aus externen Quellen.

Schlüssel zur Identifizierung einer Ereignisreihe

Ein- und Ausgabeblöcke identifizieren eine Reihe von Ereignissen, indem sie einen Schlüssel für die Reihe (oder Datenstrom) der Ereignisse festlegen. Diese Ereignisreihe dient dazu, die korrekten Ereignisse zu identifizieren, die an einen Eingabeblock übergeben werden sollen. Der Schlüssel besteht aus mehreren Blockparametern und identifiziert diese Ereignisreihe, die sich von anderen Ereignisreihen durch denselben Blocktyp unterscheidet. Zum Beispiel:

- Bei Ein- und Ausgaben von `Measurement`-Objekten besteht der Schlüssel aus Gerät, Fragment und Serie. Der **Unit**-Parameter, der in einem Ausgabeblock angegeben wird, ist nicht Bestandteil des Schlüssels (er dient nur zur Information), und es ist nicht erforderlich, dass er mit den Parametern des **Messwerteingabe**-Blocks übereinstimmt.
- Bei `Event`-Objekten besteht der Schlüssel aus Gerät und Ereignistyp.
- Bei `Alarm`-Objekten besteht der Schlüssel aus Gerät und Alarmtyp.

Einschränkungen für Ausgabeblöcke

Für synchrone Ausgabetypen wie Messwerte, Ereignisse und Alarme (siehe auch „Ausgabeblöcke und Ereignissteuerung“ auf Seite 113) ist es in Analytics Builder nicht erlaubt, mehr als einen Ausgabeblock zu haben, der Ereignisse mit einem bestimmten Schlüssel generiert.

Da Verbindungen zwischen den Modellen bestehen können, besteht der Hauptgrund für diese Einschränkung darin, Mehrdeutigkeiten oder Fehler zu vermeiden, die in den Eingabeblöcken bei der Verarbeitung von Ereignissen auftreten können, wenn mehrere Ausgabeblöcke (in verschiedenen Modellen) denselben Ausgabestrom zum selben Zeitpunkt generieren.

8 Details zu Werten und Blöcken

- Einleitung 118
- Werte als Repräsentationen von zeitkontinuierlichen physikalischen Größen 118
- Zeitfenster für die Ausgabesteuerung eines Blocks 125
- Pulse-Signale 128
- Messungen von diskreten Zeitwerten 130

Einleitung

Analytics Builder stellt eine Umgebung zur Verfügung, in der Blöcke so miteinander verbunden werden können, dass sie Modelle bilden, welche in der Lage sind, Eingaben zu verarbeiten und darauf zu reagieren. Analytics Builder verwendet intern eine Reihe von Werttypen, und es ist wichtig, die Unterschiede zwischen diesen Typen zu verstehen. In den nachfolgenden Abschnitten werden die Unterschiede zwischen den Werttypen beschrieben, die kontinuierliche Zeitwerte und diskrete Zeitwerte darstellen. Außerdem finden Sie Informationen zum `pulse`-Typ, zu Zeitfenstern bei der Implementierung von Blöcken und wann Blöcke Ausgaben generieren.

Zusammenfassung

Bei Analytics Builder werden die Werttypen `float`, `boolean` und `string` zur Darstellung kontinuierlicher Zeitwerte verwendet. Sie haben die folgenden Eigenschaften, die Sie beim Erstellen von Modellen oder benutzerdefinierten Blöcken berücksichtigen sollten:

- Eine Verbindung behält ihren Wert solange bis eine neue Eingabe erfolgt.
- Wiederholte Eingaben desselben Werts sollten sich (in den meisten Fällen) nicht auf die Ausgabe auswirken.
- Es gibt keine Garantie dafür, dass eine neue Eingabe innerhalb eines definierten Zeitraums erfolgt.

Vergleichen Sie dies mit dem `pulse`-Typ, welcher diskrete Ereignisse darstellt und die folgenden Eigenschaften aufweist:

- Ein Pulse-Signal steht für einen einzigen Zeitpunkt.
- Mehrere Eingaben eines Pulse-Signals sind von Bedeutung, auch wenn es keinen Unterschied zu einem dem Signal zugeordneten Wert gibt.
- Wenn ein Block mehrere Eingangsports für Pulse-Signale hat, dann werden Eingaben, die zu unterschiedlichen Zeiten an unterschiedlichen Ports erfolgen nur jeweils einzeln „gesehen“. Ein Eingangsport für ein Signal an einen Block verhält sich so, als ob er nach der Auswertung automatisch „zurückgesetzt“ wird.

Diese Eigenschaften und die dahinter stehenden Überlegungen werden in den nachfolgenden Abschnitten untersucht und erläutert. Dort wird auch erklärt, wie man mit Fällen umgeht, die nicht in diese Unterscheidungen passen, wie zum Beispiel diskrete numerische Messungen.

Werte als Repräsentationen von zeitkontinuierlichen physikalischen Größen

Ein zeitkontinuierlicher Werttyp, insbesondere vom Typ `float` (das heißt, numerisch), wird in der Regel dazu verwendet, die Messung einer kontinuierlichen physikalischen Größe oder Eigenschaft durch einen Sensor darzustellen. Ein Wert kann zum Beispiel Folgendes darstellen:

- Den Druck in einer Leitung.
- Die mit einem Thermometer gemessene Temperatur.

- Die Drehzahl einer Achse.
- Die Position eines Objekts.

Dies sind alles kontinuierlich messbare Eigenschaften, die analoger Natur sind. Es wird einen gewissen Grad an Präzision geben in Bezug darauf, wie genau sie sowohl in Zeit als auch Wert gemessen werden können (und innerhalb physikalischer Grenzen). Mit zeitlicher Genauigkeit meinen wir, wie häufig eine Messung durchgeführt werden kann und wie präzise der Zeitpunkt der Messung erfasst wird. Es kann auch eine Latenz geben - eine Verzögerung zwischen einer Änderung der tatsächlichen Eigenschaft und wann sie gemessen werden kann. Mit Wertgenauigkeit meinen wir, mit welchem Präzisionsgrad der Wert gemessen werden kann - typischerweise können mindestens 2 signifikante Zahlen, und selten mehr als 4 oder 5 signifikante Zahlen der Genauigkeit, unterschieden werden kann. Mit kontinuierlich meinen wir, dass es gültig ist, die Eigenschaft zu einem beliebigen Zeitpunkt zu messen.

Wenn diskrete Messungen nicht kontinuierlich, sondern zu verschiedenen Zeitpunkten vorgenommen werden, kann man dies als „Abtastung“ (siehe auch https://de.wikipedia.org/wiki/Abtastung_(Signalverarbeitung)) oder auch „Probe“ bezeichnen, und die Grenzen der Wertgenauigkeit werden als „Quantisierungsabweichungen“ bezeichnet (siehe auch https://de.wikipedia.org/wiki/Quantisierung_(Signalverarbeitung)). Bei der Messung eines kontinuierlichen Wertes sollte die Rate, mit der die Messungen erzielt werden, keinen signifikanten Unterschied bei der Ausgabe eines Blocks oder eines Modells aufweisen. Mehr Messungen können zu einer genaueren Ausgabe führen, sie sollten aber keine grobe Änderung der Berechnungen vornehmen.

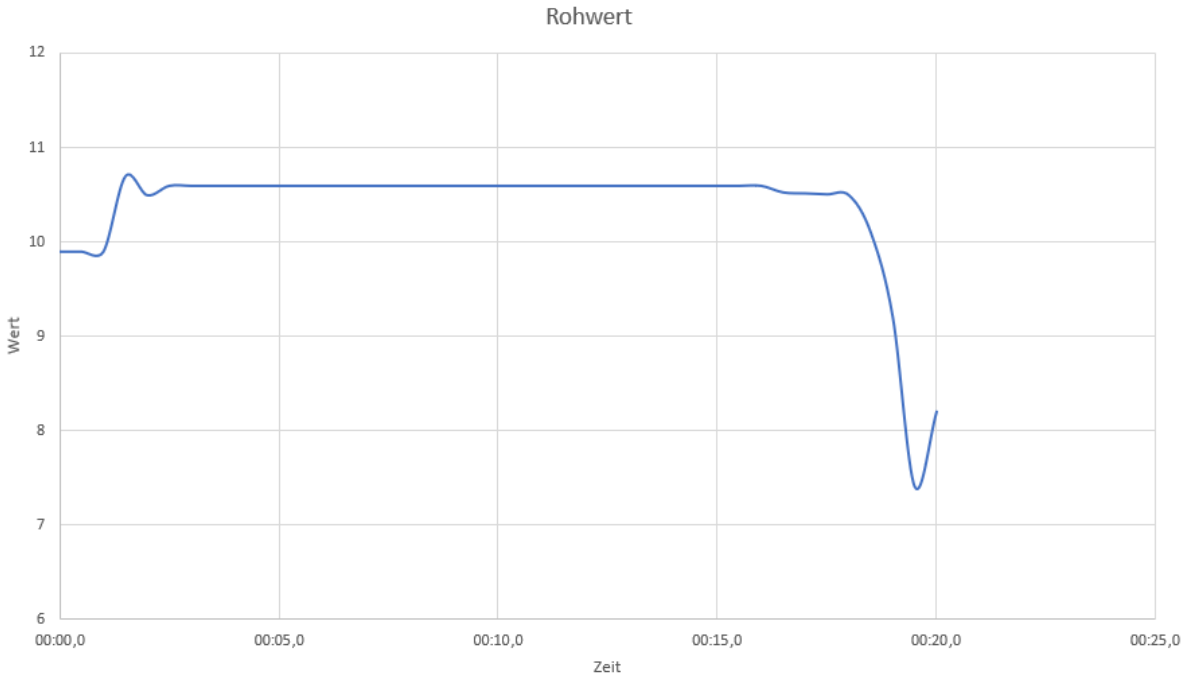
Beispiel: ein Sensor, der die Drehzahl einer Achse misst, kann jede Zehntelsekunde (0,1) einen neuen Messwert liefern und im Bereich von 0 bis 10000 U/min nur bis zu den nächsten 50 U/min messen. Eine Änderung von 10 oder 20 U/min führt möglicherweise nicht zu einer Änderung des Messwerts, da die Änderung geringer ist als der Genauigkeitsgrad. Das Anziehen von Bremsen zum Anhalten einer rotierenden Achse wird möglicherweise nicht sofort erkannt und führt zunächst zu einem Messwert von 1000 U/min, dem 0,1 Sekunden später nach dem Anhalten der Achse ein Messwert von 0 U/min folgt (wobei die Achse einige Millisekunden zum Anhalten braucht und sich in diesem Zeitraum verlangsamt).

Ein Sensor kann so angeschlossen werden, dass er eine neue Messung mit einer regelmäßigen Frequenz liefert (zum Beispiel, Audio-Sampling mit 8.000 Hz oder eine Kamera, die Videos mit 50 Bildern pro Sekunde aufnimmt). Dies ist eine regelmäßige *Abtasteingabe* (oder auch *Probeeingabe* genannt).

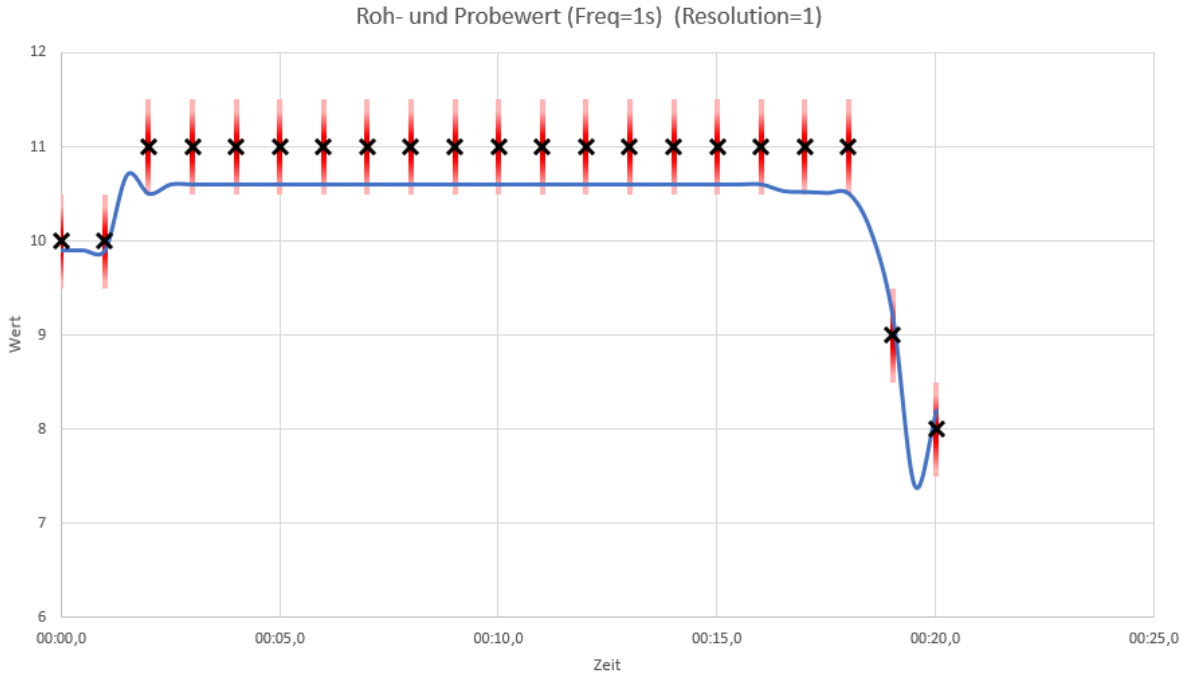
Eine einfache und übliche Optimierung besteht darin, dass ein Sensor oder ein Gerät nur dann einen neuen Messwert generieren darf, wenn sich der Wert vom vorherigen Wert unterscheidet. Für viele Sensoren wäre es normal, etwas zu messen, das einen stetigen oder konstanten Wert häufig beibehält (zumindest innerhalb der Quantisierungsgrenze), und es hat keinen Mehrwert, immer wieder denselben Wert zu senden. Dies ist eine *Änderungseingabe*. Die zugrundeliegende Abtastfrequenz bleibt bestehen, aber neue Werte werden nur dann vom Sensor übertragen, wenn sie unterschiedlich sind.

Es ist auch möglich, die regulären Formen für die Probeeingabe und die Änderungseingabe miteinander zu kombinieren: ein Sensor, der eine neue Eingabe erzeugt, wenn sich der Messwert ändert, oder periodisch. Dies ist eine *hybride Eingabe*. Beispiel: der oben beschriebene Drehsensor kann nur dann einen Wert senden, wenn sich die Drehzahl ändert, oder unabhängig davon alle 10 Sekunden.

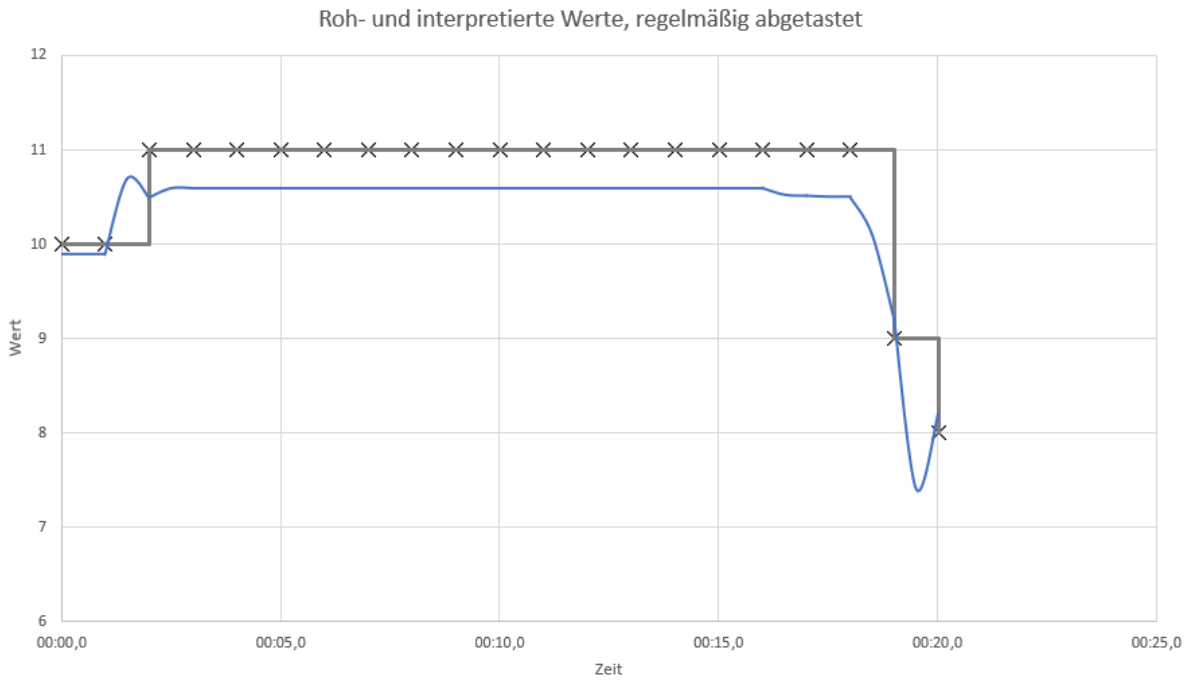
Betrachten Sie als Beispiel einen Rohwert, der sich im Laufe der Zeit wie folgt ändert:



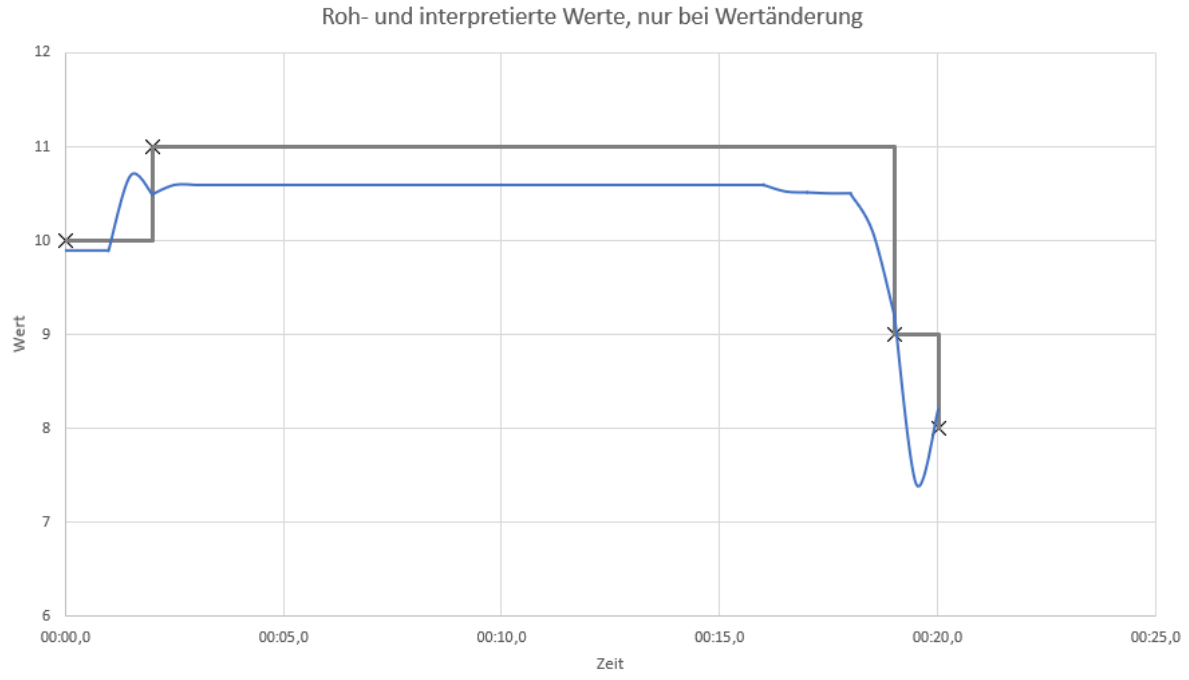
Aber angenommen, der Sensor kann nur bis zur nächsten ganzen Zahl messen, und nur einmal pro Sekunde. Der Wert hat also einen gewissen Fehler, was durch die roten Fehlerbalken angezeigt wird:



Ein regelmäßiger Abtastsensor würde einheitliche Eingaben generieren:



Während ein Änderungssensor nur dann Eingaben generiert, wenn sich ein Wert ändert:



Die graue Linie zeigt, wie ein Echtzeit-Verarbeitungssystem wie Apama solche Werte interpretiert. Es wird davon ausgegangen, dass ein Wert den letzten Wert beibehält, bis er durch einen neueren Wert ersetzt wird. Es ist auch üblich, Linien zwischen den Messungen zu zeichnen. So gibt es eine gerade Linie, die sich von Wert 11 um 00:01 Uhr auf den Wert 9 um 00:19 Uhr verringert. Ein Echtzeitsystem kann dies jedoch nicht tun. Es kann nicht wissen, was der nächste Wert sein wird, wobei die Betrachtung historischer Daten zwischen den Werten interpolieren kann. Zur Zeit 00:19,5

hat es nur die Information, dass der Wert 11 und dann 9 war. Es weiß noch nicht, dass der Wert zur Zeit 00:20 dann 8 sein wird. Beachten Sie, dass es keinen Unterschied zwischen der interpretierten grauen Linie im Fall einer regelmäßigen Abtastung und im Änderungsfall gibt. Beachten Sie, dass in der Mitte des Diagramms eine erheblicher Quantisierungsabweichung vorliegt (der `true`-Wert von 10,6 wird als 11 gelesen), und die Abtasthäufigkeit von nur einmal pro Sekunde bedeutet, dass der Mindestpunkt von 7,4 bei 00:19,5 Sekunden verloren geht.

Eingabewerte zu verschiedenen Zeiten

Stellen Sie sich zwei Positionssensoren vor, welche die Positionen von zwei Roboterarmen ausgeben, wobei beide *Änderungs*-Sensoren sind. Wenn sich die beiden Arme gleichzeitig in dieselbe Richtung und mit derselben Geschwindigkeit bewegen, dann sollten die Positionssensoren gleichzeitig auf die neuen Werte aktualisiert werden, so dass sie einen konstanten Abstand voneinander haben (oder zumindest einen annähernd konstanten Abstand). Wenn der **Differenz**-Block Eingangsports hat, die mit beiden Sensoren verbunden sind, dann sollte die Ausgabe des **Differenz**-Blocks, auch wenn sich die Roboterarme bewegen, annähernd konstant sein. Analytics Builder wertet alle Werte mit demselben Zeitstempel aus. Auch wenn es eine kleine Verzögerung beim Empfang der Werte von den beiden Sensoren geben kann (vorausgesetzt, sie liefern Zeitstempel von derselben Uhr und der **Zeitstempel ignorieren**-Parameter des **Messwert-Eingabe**-Blocks ist nicht gesetzt), dann generiert der **Differenz**-Block immer eine synchronisierte Ausgabe, wie in der folgenden Tabelle dargestellt:

Zeit	Positionssensor 1	Positionssensor 2	Ausgabe des Differenz-Blocks
00:00	4	14	10
00:01	6	16	10
00:02	9	19	10

Stellen Sie sich im Gegensatz dazu vor, dass sich die beiden Roboterarme nicht im Einklang bewegen. Zuerst bewegt sich der eine Arm, dann der andere. Während sich die Arme bewegen, kann der Abstand zwischen den Armen variieren. Da es sich bei den Sensoren um *Änderungseingaben* handelt, erhält man nur dann einen neuen Wert, wenn sich die Position ändert. Das Fehlen einer Eingabe bedeutet jedoch nicht, dass der entsprechende Roboterarm keine Position hat. Der Arm ist geblieben, wo er war (innerhalb der Fehlergrenzen). Zum Beispiel:

Zeit	Positionssensor 1	Positionssensor 2	Ausgabe des Differenz-Blocks
00:00	9	19	10
00:01	11	**19**	8
00:02	13	**19**	6
00:05	**13**	18	5
00:06	**13**	17	4

Zeit	Positionssensor 1	Positionssensor 2	Ausgabe des **Differenz**-Blocks
00:07	**13**	15	2

Die fettgedruckten Zahlen geben den effektiven Wert an. Der letzte Wert bleibt erhalten, wenn er nicht durch einen neueren Wert ersetzt wurde.

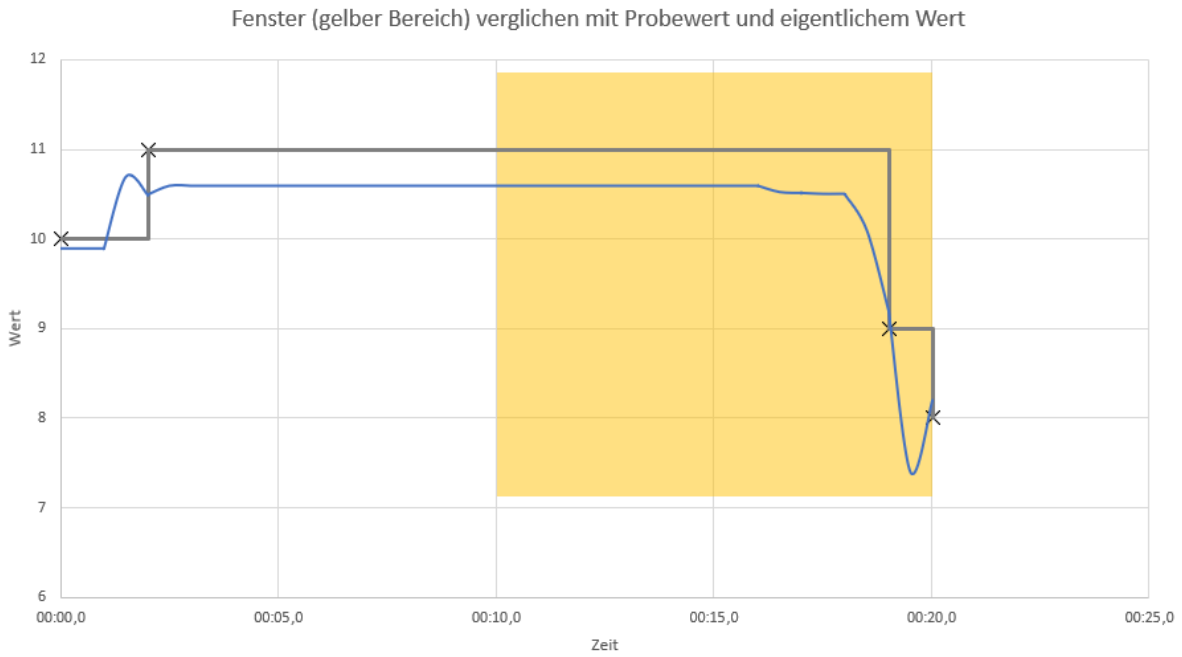
Änderungseingaben und Zeitfenster

Wenn eine Änderungseingabe mit einem Aggregatblock wie zum Beispiel dem **Durchschnitt (Mittelwert)**-Block verbunden ist, dann sollte der Block die Eingabe so behandeln, als ob der Block kontinuierlich den neuesten Wert empfängt. Dies ist bei den Blöcken wichtig, die ein Zeitfenster verwalten. Selbst wenn der Block zuletzt eine Eingabe erhalten hat (und dadurch seine `$process`-Aktion aufgerufen wurde), die länger als das Zeitfenster zurückliegt, dann enthält das Fenster den neuesten Wert. Beispiel: bei den Blöcken **Durchschnitt (Mittelwert)** und **Integral** ist die Fensterdauer auf 10 Sekunden eingestellt ist und die Eingabe erfolgt so:

Zeit	Eingabewert	Fensterinhalt	Ausgabe des **Durchschnitt (Mittelwert)**-Blocks	Ausgabe des **Integral**-Blocks
00:00	10	0: 10	10	0
00:02	11	0-2: 10	10	20
00:10	**11**	0-2: 10, 2-10: 11	10.8	108
00:12	**11**	2-12: 11	11	110
00:19	9	9-19: 11	11	110
00:20	8	10-19: 11; 19-20: 9	10.8	108

Beachten Sie in diesem Fall, wie ein zur Zeit 00:02 empfangener Messwert noch Einfluss auf die Ausgabe um 00:19 und später hat - weil er erst um 00:19 ersetzt wird. Beachten Sie auch, dass wenn ein neuer Wert auftritt, dieser überhaupt keinen Einfluss auf den Durchschnitt oder das Integral hat - er hatte noch zu keinem Zeitpunkt diesen Wert. Die einzige Ausnahme ist der **Durchschnitt (Mittelwert)**-Block, wenn er startet - bei einem leeren Fenster ist der Ausgabewert mit dem Eingabewert identisch.

Sehen Sie auch im Diagramm unten, welche Werte das Fenster zur Zeit 00:20 abdeckt:



Während innerhalb des Fensters nur die aktualisierten Messungen mit den Werten 9 und 8 empfangen wurden, liegt der Durchschnittswert innerhalb des Fensters nahe bei 11. Die aktualisierte Messung dafür wurde zur Zeit 00:02 empfangen, aber da es sich um einen kontinuierlichen Wert handelt, wird der Wert 11 bis zur Zeit 00:19 beibehalten.

Beachten Sie, dass die Abwesenheit von Eingaben über einen längeren Zeitraum bei einem Block wie **Fehlende Daten** das Verhalten des Blocks beeinflussen kann. Wäre der **Fehlende Daten**-Block mit einer Dauer von 10 Sekunden konfiguriert ist, dann würde er zur Zeit 00:12 auslösen.

Wenn die Blöcke **Durchschnitt (Mittelwert)** und **Integral** regelmäßige Eingaben von einem regulären Probesensor erhalten, dann erhält der Block mehr Messwerte, und die vergleichbare Tabelle sieht so aus:

Zeit	Eingabewert	Fensterinhalt	Ausgabe des Durchschnitt (Mittelwert)-Blocks	Ausgabe des Integral-Blocks
00:00	**10**	**0: 10**	**10**	**0**
00:01	10	0-1: 10	10	10
00:02	**11**	**0-2: 10**	**10**	**20**
00:03	11	0-2: 10, 2-3: 11	10.333	31
00:04	11	0-2: 10, 2-4: 11	10.5	42
00:05	11	0-2: 10, 2-5: 11	10.6	53
00:06	11	0-2: 10, 2-6: 11	10.667	64

Zeit	Eingabewert	Fensterinhalt	Ausgabe des Durchschnitt (Mittelwert)-Blocks	Ausgabe des Integral-Blocks
00:07	11	0-2: 10, 2-7: 11	10.714	75
00:08	11	0-2: 10, 2-8: 11	10.75	86
00:09	11	0-2: 10, 2-9: 11	10.778	97
00:10	**11**	**0-2: 10, 2-10: 11**	**10.8**	**108**
00.11	11	1-2: 10, 2-11: 11	10.9	109
00:12	**11**	**2-12: 11**	**11**	**110**
00:13	11	3-13: 11	11	110
00:14	11	4-14: 1	11	110
00:15	11	5-15: 11	11	110
00:16	11	6-16: 11	11	110
00:17	11	7-17: 11	11	110
00:18	11	8-18: 11	11	110
00:19	**9**	**9-19: 11**	**11**	**110**
00:20	**8**	**10-19: 11; 19-20: 9**	**10.8**	**108**

Beachten Sie, dass die hervorgehobenen Zeilen die gleichen sind wie ohne die wiederholten Messungen. Wiederholte Messungen desselben Wertes, die von diesen Blöcken empfangen werden, machen keinen Unterschied zu dem, was der Block bei einer erneuten Auswertung berechnen würde.

Zeitfenster für die Ausgabesteuerung eines Blocks

Bei Aggregatblöcken wie dem **Durchschnitt (Mittelwert)**-Block bewirkt eine Änderung des Eingabewerts, dass sich die Ausgabe des Blocks bei regelmäßiger Neuauswertung ändert und sich dem neuen Wert annähert. Wenn der Block in der Vergangenheit verschiedene Eingabewerte erhalten hat, dann ist eine Neuauswertung des Blocks zu jedem beliebigen Zeitpunkt möglich, und jede Neuauswertung kann eine andere Ausgabe generieren.

Mit dem vorherigen Beispiel aus „Änderungseingaben und Zeitfenster“ auf Seite 123 ergibt die wiederholte Neuauswertung eines **Durchschnitt (Mittelwert)**-Blocks in einem 10-Sekunden-Fenster die folgenden Werte:

Zeit	Eingabewert	Ausgabe des **Durchschnitt (Mittelwert)**-Blocks
00:00	10	10
00:01	10	10
00:02	11	10
00:03	11	10.333
00:04	11	10.5
00:05	11	10.6
00:06	11	10.667
00:07	11	10.714
00:08	11	10.75
00:09	11	10.778
00:10	11	10.8
00.11	11	10.9
00:12	11	11

Es wäre auch möglich, den Block jede halbe Sekunde oder jeden Bruchteil einer Sekunde neu auszuwerten - auf Millisekunden oder noch kleinere Zeiten zwischen den Neuberechnungen. Es ist unpraktisch, den Block „kontinuierlich“ neu auszuwerten (um den Durchschnittswert zu einem bestimmten Zeitpunkt neu zu berechnen und eine neue Ausgabe zu generieren). Wann ist also ein geeigneter Zeitpunkt für die Auswertung des Blocks und das Generieren einer Ausgabe?

Der **Durchschnitt (Mittelwert)**-Block hat den Parameter **Ausgabe-Schwellenwert**. Einige andere Blöcke haben diesen Parameter auch. Wenn dieser Parameter gesetzt ist, dann emuliert der Block einen Sensor, der immer dann einen neuen Messwert erzeugt, wenn sich die Ausgabe um den Ausgabe-Schwellenwert ändert. Wenn er also auf 0,1 eingestellt ist, dann erhalten wir mehrere Ausgaben zwischen 00:02 und 00:03 (wenn sich die Ausgabe von 10 auf 10,333 ändert), eine weitere Ausgabe zwischen 00:03 und 00:04 (wenn er 10,4 erreicht), Ausgaben genau um 00:04 und 00:05, eine weitere zwischen 00:06 und 00:07 (10,7), und dann um 00:10 und 00:11 und 00:12. Der Block berechnet, zu welchem Zeitpunkt sich die Ausgabe im Vergleich zur letzten Ausgabe um mehr als den Ausgabe-Schwellenwert ändern würde, und wertet zu diesem Zeitpunkt neu aus. Zeitlich kann die Ausgabe also recht unregelmäßig erfolgen, sie kann aber zu solchen Zeitpunkten erfolgen, dass die Ausgabe immer um den Wert abweicht, der dem Ausgabe-Schwellenwert entspricht. Der Block wertet auch bei allen neuen Eingaben neu aus, auch wenn sich der Wert nicht von der letzten Eingabe unterscheidet.

Damit die Modelle zu einem beliebigen Zeitpunkt Berechnungen mit der Ausgabe des **Durchschnitt (Mittelwert)**-Blocks durchführen können (zum Beispiel für den Vergleich mit einer anderen Messung), ist auch ein **Probe**-Eingangsport vorhanden, mit dem eine Neuauswertung erzwungen und eine neue Ausgabe generiert werden kann.

Zeitfenster und Buckets

Eine Reihe von Blöcken, hauptsächlich die in der **Aggregat**-Kategorie, verwalten ein zeitbasiertes Fenster mit Eingabewerten, die in der Vergangenheit empfangen wurden. Die Ausgabe dieser Blöcke ist eine Berechnung, die auf den Werten in diesem Fenster basiert. In der Regel bieten solche Blöcke zwei verschiedene Möglichkeiten, dieses Fenster zu verwalten:

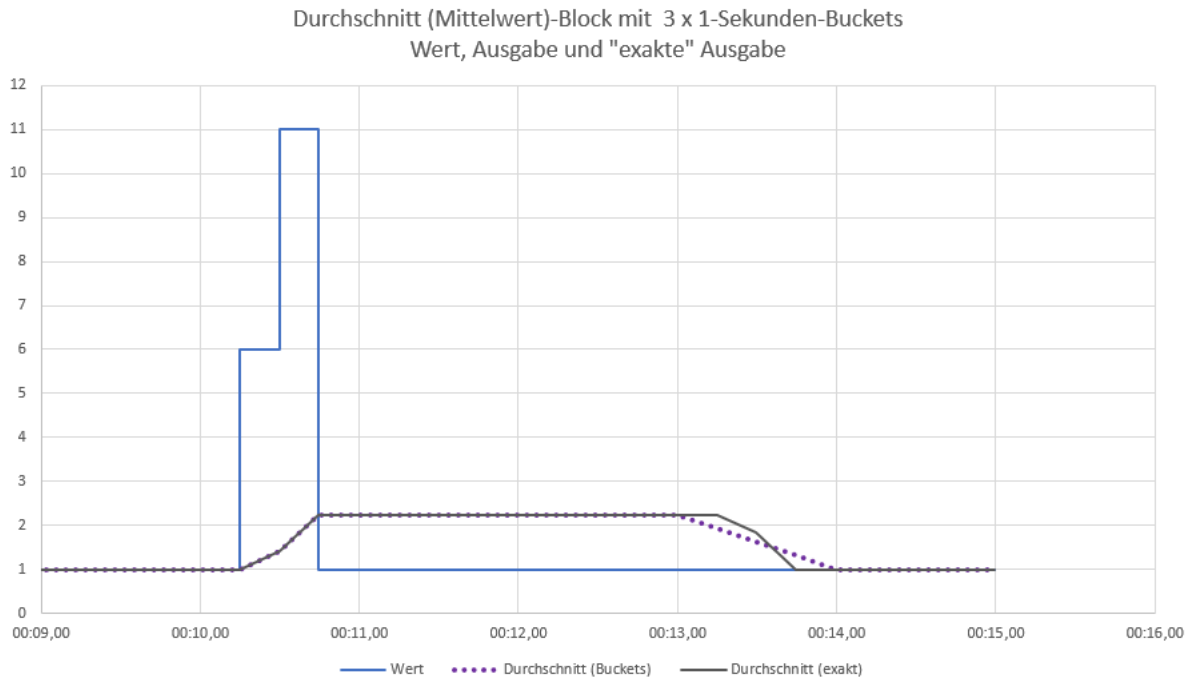
- Ein Parameterwert, der die Dauer des Fensters angibt. Wenn er gesetzt ist, dann lässt das Fenster automatisch die interpretierten Werte verfallen, die älter als die angegebene Zeit sind (wobei die interpretierten Werte der letzte empfangene Wert zu einem beliebigen Zeitpunkt ist, wie in „Änderungseingaben und Zeitfenster“ auf Seite 123 beschrieben). Wenn der Parameter nicht angegeben ist, dann lässt der Block die Daten nicht automatisch verfallen.
- Ein Eingangsport zum Zurücksetzen. Wenn ein Signal empfangen wird, dann wird der Inhalt des Fensters gelöscht und der Block wird so zurückgesetzt, so dass er keinen Inhalt mehr hat.

Man kann diese Möglichkeiten kombinieren, oder keine davon verwenden. Aber typischerweise benutzt man entweder die eine oder die andere Möglichkeit.

Für den Fall, dass die Fensterdauer angegeben wird, muss der Block in der Lage sein, alte Daten verfallen zu lassen. Eine streng exakte Implementierung wäre, jede abweichende Messeingabe zu speichern, zusammen mit dem Zeitpunkt seines Auftretens. Bei langen Fenstern und/oder hochfrequenten Eingaben kann dies dazu führen, dass eine große Menge an Daten gespeichert wird. Um eine übermäßige Datenspeicherung zu vermeiden, werden von den Produktblöcken nicht alle Messwerte und Zeiten gespeichert. Stattdessen wird die Fensterdauer in gleich große Buckets unterteilt. Die Blöcke speichern den Status pro Bucket und verwenden diese Informationen, um die Ausgabe des Blocks neu zu berechnen. Ein historischer Bucket kann entweder vollständig im Fenster liegen oder teilweise verfallen sein. Wenn ein Bucket teilweise verfallen ist, dann wendet der Block einen Bruchteil der Werte innerhalb dieses Bereichs an. Die praktische Auswirkung davon ist: wenn sich der Wert ohne signifikante Fluktuation ändert, dann gibt es nur einen kleinen Unterschied zwischen einer exakten (aber ressourcenintensiveren) Implementierung des Blocks und einer, die Buckets verwendet.

Wenn es eine signifikante Fluktuation im Eingabewert gibt, die eine Verschiebung der Ausgabe verursacht, dann geht die genaue Zeit der einzelnen Messeingaben verloren, und die Auswirkung, die der signifikante Wert bei seinem Verfall hat, wird zeitlich um bis zu einer Bucket-Dauer verteilt. Die Produktblöcke verwenden 20 Buckets als vernünftigen Kompromiss zwischen Genauigkeit und Effizienz.

Um dies zu veranschaulichen, übertreiben wir den Effekt, indem wir einen **Durchschnitt (Mittelwert)**-Block mit 3 Buckets und einem 3-Sekunden-Fenster simulieren, so dass jeder Bucket eine Dauer von 1 Sekunde hat. Einige anomale Messwerte (nach einer kontinuierlichen Eingabe des Wertes 1) beeinflussen den Durchschnitt sowohl für exakte als auch Bucket-basierte **Durchschnitt (Mittelwert)**-Blöcke auf die gleiche Weise, aber wir können das Ergebnis des „exakten“ Verfallens jedes Werts genau 3 Sekunden nach seinem Auftreten mit der Verwendung von Buckets vergleichen, bei denen die Änderung in der Ausgabe über die Bucket-Dauer geglättet wird:



Beachten Sie, dass nicht nur der Zeitpunkt des Verfalls der anomalen Werte weniger präzise ist, sondern dass auch die exakte Form der Ausgabe verloren geht. Der Bucket-Durchschnitt ändert sich gleichmäßig zwischen 00:13 und 00:14. Denken Sie daran, dass die Produktblöcke 20 Buckets verwenden, so dass der Effekt in diesem Fall weniger stark ausgeprägt wäre.

Pulse-Signale

Ein Pulse-Signal wird verwendet, um einen Zeitpunkt oder eine Zustandsänderung anzuzeigen. Beispiele für Anwendungsfälle von Pulse-Signalen sind:

- Eine Person geht durch eine Pforte (zum Beispiel, in einem Bahnhof).
- Eine Taste wird gedrückt (zum Beispiel, eine Notstopp-Taste).
- Eine Maschine geht in einen neuen Zustand über (zum Beispiel, ein Gateway wird zurückgesetzt oder eingeschaltet).
- Ein Gerät hat eine Verbindung zum Netzwerk hergestellt.

In Cumulocity IoT werden Ereignisse, Alarme und Operationen als Quellen für Pulse-Signale verwendet. Siehe auch die folgenden Abschnitte in der englischsprachigen Dokumentation von Cumulocity IoT: Events und Operations im *Concepts guide* und Alarms im *Reference guide*.

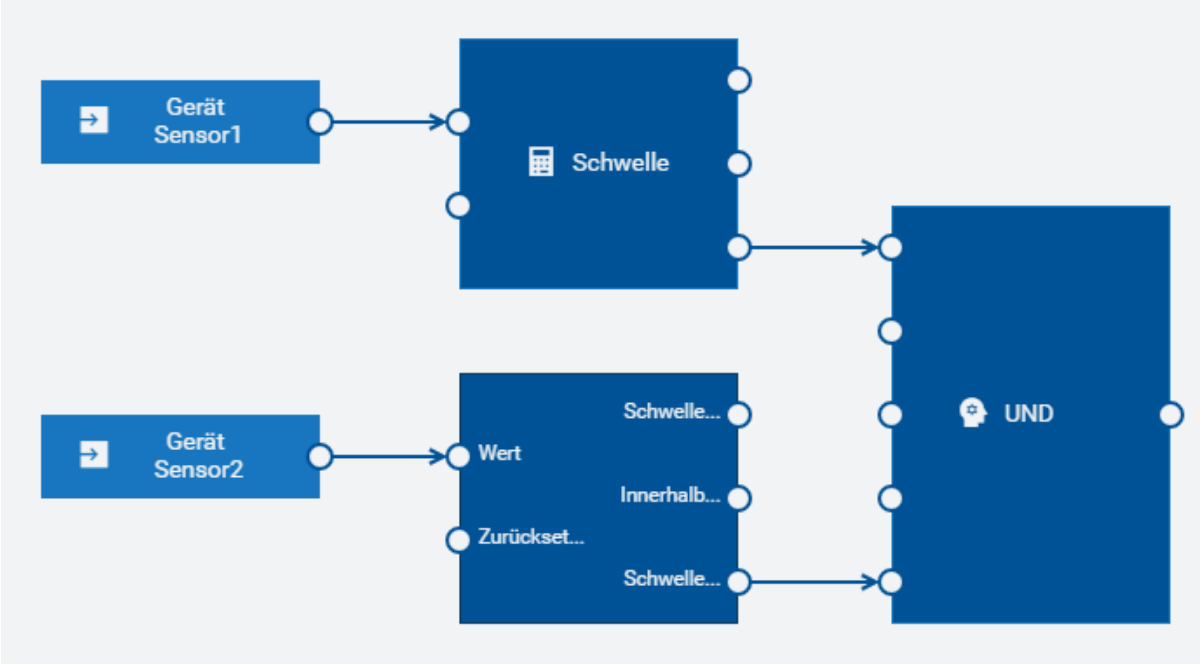
Ein Pulse-Signal kann einfach nur ein Zeitpunkt sein, es kann aber auch zusätzliche Informationen enthalten, zum Beispiel, die Versionsnummer der Software oder mit welchem Netzwerkknoten sie sich verbunden hat. Diese Informationen können mit dem **Eigenschaft extrahieren**-Block abgerufen werden. Wenn Sie Ihre eigenen benutzerdefinierten Blöcke schreiben, dann sind diese Informationen verfügbar, wenn die Eingabe als ein `Value`-Typ deklariert ist, der ein Eigenschaftsfeld hat. Dies kann auch mit numerischen Werttypen verwendet werden. Weitere Informationen über den `Value`-Typ finden Sie in der Dokumentation für das Analytics Builder Block SDK.

Im Gegensatz zu Messwerten ist das Timing und die Anzahl der Pulse-Signale signifikant, und auch wenn der einzige Unterschied zwischen aufeinanderfolgenden Signalen der Zeitpunkt des Empfangs sein kann, so ist jeder einzelne dennoch signifikant (während mehrere Messungen mit demselben Wert von geringem Interesse sind).

Im Gegensatz zu Messwerten ist ein Pulse-Signal nur für eine einzige Auswertung eines Modells aktiv, wobei eine Modellauswertung alle Blöcke verarbeitet, die über einen Timer verfügen, der ausgelöst wird (einschließlich Eingabeblöcke) und alle Blöcke, die mit Ausgangsports verbunden sind, bei denen sich der Wert geändert hat. Während sowohl Pulse-Signale als auch boolesche Messwerte durch den `boolean`-Typ im Block-EPL dargestellt werden, ist ihr Verhalten dennoch unterschiedlich:

- Wenn ein boolescher Messwert von einem Block empfangen wird, dann bleibt er solange bei diesem Wert, bis er ersetzt wird. Wenn Sie zum Beispiel einen **UND**-Block benutzen, um zu untersuchen, ob die Temperatursensoren 1 und 2 beide über einem Schwellenwert liegen, dann ist dieser Wert auch nach dem Empfang hoher Messwerte noch wahr.
- Wenn ein Pulse-Signal von einem Block empfangen wird, dann wird der Block nach der Signalauswertung zurückgesetzt. Wenn zum Beispiel ein **Durchschnitt (Mittelwert)**-Block durch ein Ereignis zurückgesetzt wird, dann erfolgt das Zurücksetzen beim Empfang des Ereignisses. Wenn danach kein weiteres Ereignis mehr empfangen wird, dann wird der Block bei zukünftigen Werteingaben nicht zurückgesetzt.

Es ist immer noch gültig und sinnvoll, mehrere Pulse-Signale zu kombinieren, zum Beispiel mit einem **UND-**Block. Wenn zwei Signale zum selben Zeitpunkt auftreten, dann ist dies eine einzige Auswertung. Der **Schwelle**-Block hat zum Beispiel den **Schwelle überschritten**-Ausgangsport, der nur dann ein Signal sendet, wenn eine kontinuierliche Werteingabe von einer Seite der Schwelle zur anderen Seite geht. Zwei Sensoren auf demselben Gerät (also mit demselben Zeitstempel) können gleichzeitig Schwellen überschreiten, so dass das **UND** die Ausgabe von zwei solchen Schwellen nur dann auslöst, wenn beide Eingaben die Schwelle mit neuen Werten desselben Zeitstempels überschreiten. Beachten Sie, dass, wenn erst ein Sensor die Schwelle und der andere Sensor die Schwelle erst später überschreitet, die folgenden Blöcke niemals eine Ausgabe des **UND**-Blocks liefern würden. Sie würden dies nur tun, wenn beide gleichzeitig auftreten.



Messungen von diskreten Zeitwerten

Es gibt einige Fälle, bei denen ein numerischer Messwert keine zeitkontinuierliche Eigenschaft darstellen würde. Zum Beispiel:

- Das Gewicht von Paketen, die eine Waage auf einem Förderband passieren.
- Die Größe von Objekten, die eine Messstelle passieren.
- Der Wert eines Tickets, das von einer Maschine gescannt oder gedruckt wird.

Im Gegensatz zu den zeitkontinuierlichen Werten ist jeder dieser Werte signifikant, auch wenn zwei Messungen denselben Wert haben. Der Zeitpunkt jeder Messung kann eine gewisse Bedeutung haben, aber die Zeit zwischen den aufeinander folgenden Messungen ist nicht von großer Bedeutung. Würden die Messungen mit leicht unterschiedlichen Zeitpunkten, oder möglicherweise sogar nicht in der richtigen Reihenfolge, empfangen werden, würde dies keinen Unterschied machen (so ändert sich zum Beispiel die Summe der Ticketwerte nicht, wenn sie in einer anderen Reihenfolge oder mit unterschiedlichen Zeitpunkten verarbeitet werden; und es ist unwahrscheinlich, dass die Zeit zwischen den Werten einheitlich ist). Beachten Sie, dass wir uns mit zeitdiskret nur auf den *Zeitpunkt* der Messungen beziehen. Der Wert kann immer noch kontinuierlich sein. Das Gewicht ist zum Beispiel ein kontinuierlicher Wert, wir können aber einzelne Pakete wiegen - wobei das Gewicht eines Pakets in Grammbruchteilen darstellbar sein kann. Zwischen zwei Paketen auf einem Förderband gibt es keinen „aktuellen“ Wert für das Gewicht eines Pakets. Der Wert könnte auch diskret sein. Zum Beispiel ist der Ticketwert typischerweise ein diskreter Wert (zum Beispiel, auf den nächsten Cent gerundet oder einer von einigen vordefinierten Ticketwerten).

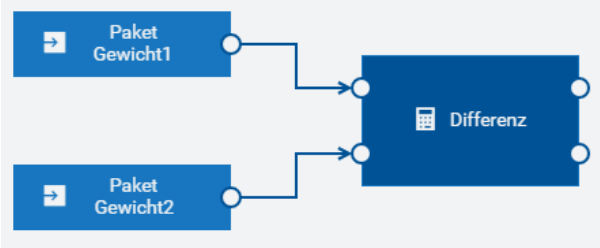
Siehe auch: https://en.wikipedia.org/wiki/Discrete_time_and_continuous_time und https://en.wikipedia.org/wiki/Continuous_or_discrete_variable. In der Praxis sind alle Messungen Stichproben einer zeitkontinuierlichen Eigenschaft.

Wenn Sie mit zeitdiskreten Eingaben arbeiten, dann sollten Sie den **Diskrete Statistik**-Block statt des **Durchschnitt (Mittelwert)**-Blocks verwenden. Es ist zwar möglich, eine Eingabe von einem Paketgewichtssensor mit dem **Durchschnitt (Mittelwert)**-Block zu verbinden, aber der **Durchschnitt (Mittelwert)**-Block gewichtet nach Zeit. Beispiel:

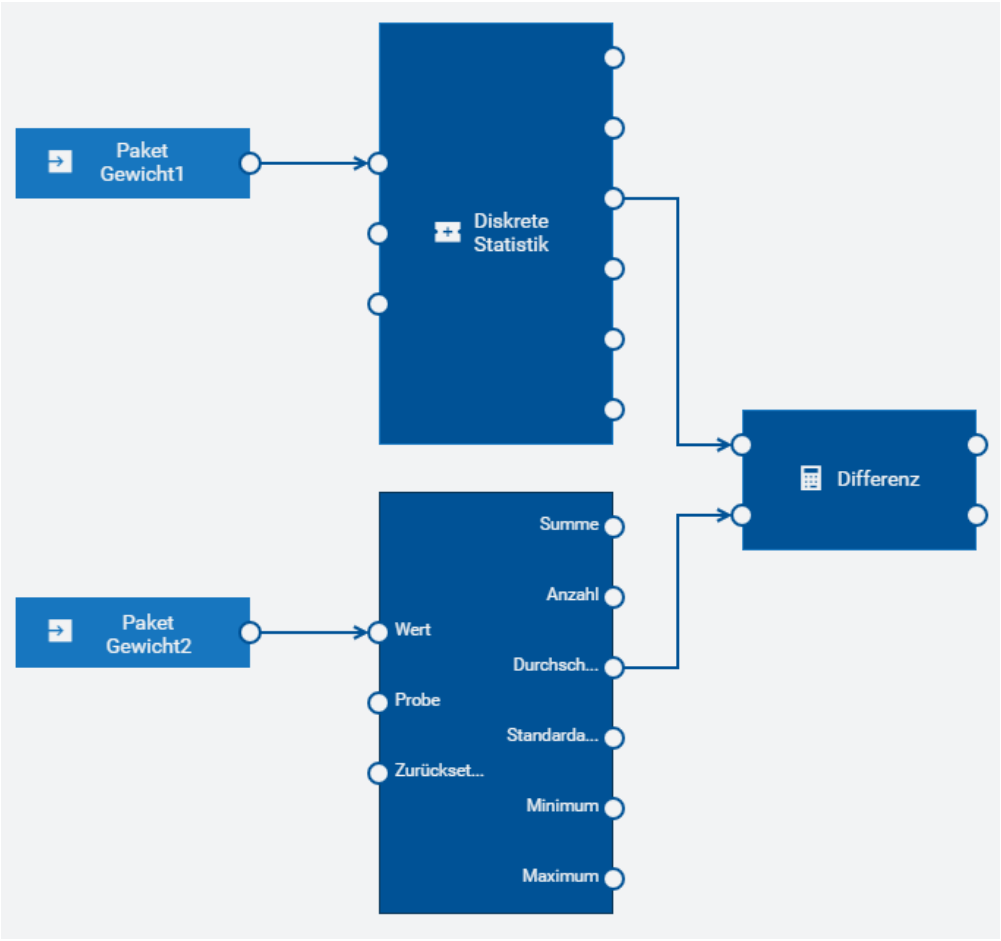
Zeit	Eingabewert	Durchschnitt bei zeitkontinierlicher Eingabe	Durchschnitt bei zeitdiskreter Eingabe
00:10	11	11	11
00:19	9	11	10
00:20	8	10.8	9.33

Vergleichen Sie dies mit der Tabelle in „Änderungseingaben und Zeitfenster“ auf Seite 123 und betrachten Sie die Zeiten ab 00:10 (das heißt, was von 00:10 bis 00:20 im Fenster wäre). Beachten Sie, dass der zeitkontinuierliche Block eine andere Ausgabe generieren würde, wenn die Eingaben zu unterschiedlichen Zeiten auftreten würden. Ein Block, der Durchschnittswerte basierend auf der diskreten Zeit ermittelt, würde das nicht tun.

Beachten Sie, dass Messungen standardmäßig als zeitkontinuierliche Werte behandelt werden. So ist es zum Beispiel möglich, die Differenz zwischen zwei Werten zu berechnen:



Das Beispiel oben gibt die Differenz zwischen dem letzten von zwei Sensoren empfangenen Gewicht an. Wenn es sich hierbei um echte zeitdiskrete Eingaben handelt, dann ist dies möglicherweise keine besonders nützliche Unterscheidung. Es kann jedoch sinnvoll sein, die Differenz der Durchschnittswerte (oder Mittelwerte) zwischen zwei zeitdiskreten Eingaben zu vergleichen. Der **Durchschnitt**-Ausgangsport des **Diskrete Statistik**-Blocks gibt einen zeitkontinuierlichen Wert aus:


Paket
Gewicht1
Diskrete
Statistik
Differenz
Summe
Anzahl
Paket
Gewicht2
Wert
Durchsch...
Probe
Standarda...
Zurückset...
Minimum
Maximum

9 Modelle und Geräte

- Modellausführung für verschiedene Geräte 134
- Broadcast-Geräte 136
- Virtuelle Geräte 137
- Verbindungen zwischen Modellen 138
- Die Anzahl der angezeigten Geräte, Gerätegruppen und/oder Assets konfigurieren ... 141
- Nach Eingabe- und Ausgabe-Assets suchen 141

Modellausführung für verschiedene Geräte

Modelle werden unabhängig voneinander ausgeführt. Das heißt, die Modelle für spezifische Geräte können parallel ausgeführt werden, wenn möglich unter Verwendung von Hardware-Parallelität, wenn die Modelle Daten (wie zum Beispiel `Measurement`-, `Event`- oder `Operation`-Objekte) für ein anderes Set von Geräten verarbeiten. Wenn Sie ein Modell definieren, dann können Sie es so konfigurieren, dass Daten von einem Set von spezifischen Geräten oder von einer Gruppe von Geräten verwendet werden, wobei jedes Gerät unabhängig behandelt wird.

Jedes Modell muss entweder:

- Eingaben von einer Reihe von spezifischen Geräten empfangen und Ausgaben an eine Reihe von spezifischen Geräten senden, oder
- Eingaben vom jedem Gerät innerhalb von Gerätegruppen empfangen und Ausgaben an das Trigger-Gerät senden.

  Eine Gerätegruppe ist ein Mittel zum Organisieren von Geräten. Eine Gerätegruppe kann beliebig viele Geräte oder auch weitere Gerätegruppen enthalten. Wenn ein Modell eine Gerätegruppe verwendet, dann wirkt sich das Modell auf alle Geräte in dieser Gerätegruppe aus, entweder direkt oder indirekt über Mitglieder der Gerätegruppe, die selbst Gerätegruppen sind und Gerätemitglieder haben (oder sogar "Enkelkinder"-Gerätegruppenmitglieder). Ein Gerät kann Mitglied von keiner, einer oder mehreren Gerätegruppen sein. Siehe auch die Informationen zur Device Management-Anwendung und dem Gruppieren von Geräten im *User guide* (hiervon gibt es auch eine deutsche Version) unter https://www.cumulocity.com/guides/.

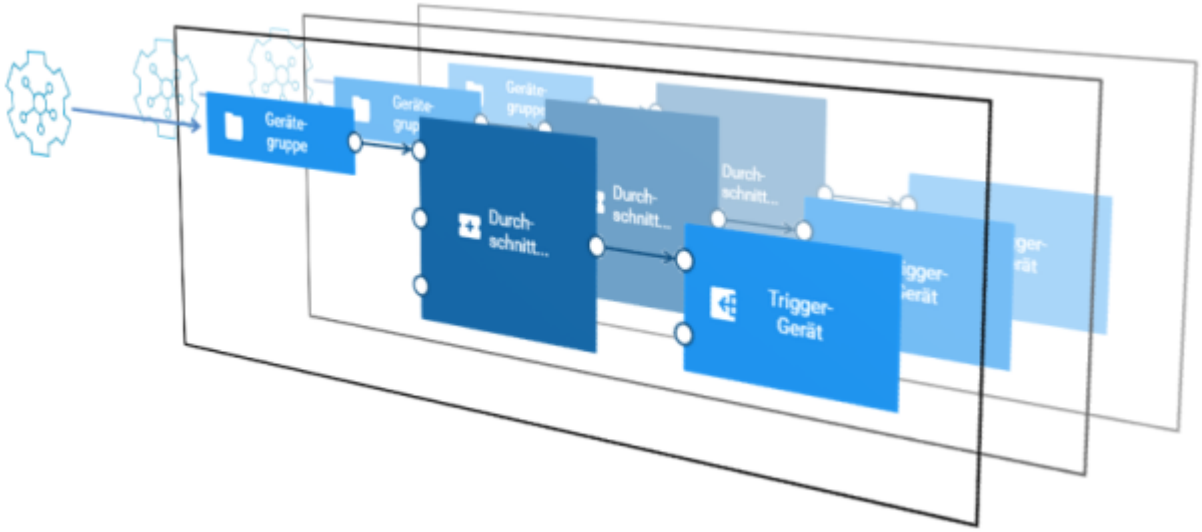
  Anmerkung:

  Ein Modell ermittelt die Mitglieder einer Gerätegruppe nur beim Aktivieren des Modells. Wenn sich die Mitglieder einer Gerätegruppe während der Ausführung des Modells ändern, dann hat dies keinen Einfluss auf das Modellverhalten; neu hinzugefügte oder entfernte Mitglieder werden bei der Ausführung nicht berücksichtigt. Wenn Sie die Mitglieder einer Gerätegruppe ändern, dann sollten Sie alle Modelle, die diese Gerätegruppe benutzen, deaktivieren und anschließend wieder aktivieren.

Es ist nicht möglich, die beiden oben genannten Eingabeblocktypen zu kombinieren (siehe jedoch „Broadcast-Geräte“ auf Seite 136). Daten von einem Modell, das spezifische Geräte verarbeitet, können jedoch an andere Modelle gesendet und von diesen empfangen werden, einschließlich Modellen für Gerätegruppen, und umgekehrt (siehe „Verbindungen zwischen Modellen“ auf Seite 138).

Wenn ein Modell Daten aus Gerätegruppen verwendet, dann verhält sich das Modell so, als würden mehrere Instanzen dieses Modells ausgeführt (wie unten veranschaulicht), wobei jede Instanz die Daten von jedem Gerät unabhängig verarbeitet. Jede Instanz verarbeitet Daten für ein anderes Gerät, aber alle haben die gleichen Blöcke und Blockparameter. Die Werte auf den Verbindungen sind bei den verschiedenen Instanzen unabhängig. Alle zustandsorientierten Blöcke, wie zum Beispiel der **Durchschnitt (Mittelwert)**-Block, arbeiten unabhängig von den Daten anderer Geräte. Wenn ein Block (wie bei Modellen mit spezifischen Geräten) einen Laufzeitfehler oder

eine Ausnahme verursacht, dann wird das gesamte Modell in einen fehlerhaften Zustand versetzt und die Verarbeitung der Daten wird für alle Geräte gestoppt.



Wenn für die Eingaben Gerätegruppen verwendet werden, dann benutzen alle Eingabeblöcke normalerweise dieselbe Gruppe. Es ist möglich, verschiedene Gerätegruppen zu verwenden. Wenn sich Geräte in einer Gruppe befinden, jedoch nicht in einer anderen, dann generieren diese Blöcke niemals ein Signal für die Geräte, die nicht in dieser Gruppe sind. Bei einigen Blöcken, wie zum Beispiel dem **Ausdruck**-Block, ist dies nicht hilfreich. Ein **Ausdruck**-Block generiert nur dann eine Ausgabe, wenn alle erforderlichen Eingänge einen Wert erhalten haben. Für die `pulse`-Eingänge eines **Gate**-Blocks kann dies jedoch hilfreich sein.

Wenn ein Modell Eingänge hat, die Daten von spezifischen Geräten verwenden, dann können die Ausgabeblöcke, welche die Ausgaben erzeugen, dasselbe Gerät oder verschiedene spezifische Geräte angeben.

Wenn ein Modell Eingänge hat, die Daten aus einer Gerätegruppe verwenden, dann müssen alle synchronen Ausgabeblöcke das **Trigger-Gerät** angeben. Dies ist ein spezieller Block, den Sie in der **Ausgabe**-Kategorie der Palette finden. Das Trigger-Gerät generiert Daten (`Measurement`, `Event` oder `Operation`) für jedes Gerät, für das die Instanz gilt - oder für jedes Gerät, das die Daten gesendet hat, um diese Instanz auszulösen. Asynchrone Ausgabeblöcke in solchen Modellen können das Trigger-Gerät oder ein spezielles Gerät angeben.

Wenn ein Vorlagenparameter für einen Ausgabeblock verwendet wird und wenn der Wert dieses Parameters eine Gerätegruppe benennt, dann wird dies so behandelt, als ob sich der Parameter auf das Trigger-Gerät bezieht. Die Ausgabe geht an das Gerät, das die Auswertung des Modells ausgelöst hat, wobei jedes Gerät innerhalb einer Gruppe unabhängig behandelt wird. In der Regel wird derselbe Vorlagenparameter sowohl für die Eingabe als auch für die Ausgabe verwendet; so beziehen sie sich auf dieselbe Gruppe und jedes Gerät wird unabhängig verarbeitet.

Mit dem Modelleditor können Sie die Geräte oder Gerätegruppen in den Eingabe- und Ausgabeblöcken durch andere ersetzen. Wenn Sie eine Gerätegruppe durch ein Gerät ersetzen, dann schalten die Ausgabeblöcke zwischen dem Trigger-Gerät und dem angegebenen Gerät um, sodass das Modell in einem einsatzfähigen Zustand bleibt. Siehe auch „Geräte, Gerätegruppen und Assets ersetzen“ auf Seite 76.

Die Test- und Simulationsmodi sind nur für Modelle mit spezifischen Geräten zulässig. Wenn Sie ein Modell mit einer Gerätegruppe testen oder simulieren möchten, dann sollten Sie es mit dem Modelleditor so abändern, dass es für ein einzelnes Gerät innerhalb der Gerätegruppe gilt und aktivieren Sie das Modell anschließend im Test- oder Simulationsmodus. Weitere Informationen zu diesen Modi finden Sie unter „Ein Modell aktivieren“ auf Seite 56.

Den Gleichzeitigkeitsgrad definieren

Standardmäßig verwendet die Analytics Builder-Laufzeit 1 CPU-Kern zur Ausführung von Modellen. Sie können die Anzahl der CPU-Kerne ändern, indem Sie eine `POST`-Anfrage an Cumulocity IoT senden, die den Wert des Keys `numWorkerThreads` ändert. Ausführliche Informationen finden Sie unter „Konfiguration“ auf Seite 157.

Normalerweise wird dieser Konfigurationswert auf die Anzahl der für das System verfügbaren CPU-Kerne gesetzt. Es kann jedoch nützlich sein, diesen Wert je nach den verfügbaren Ressourcen höher oder niedriger zu konfigurieren. Er muss nicht auf die Anzahl der Geräte skaliert werden (das heißt, es ist durchaus sinnvoll, 4 Worker-Threads mit Hunderten von Geräten zu haben, wenn man von einer moderaten Ereignisrate pro Gerät ausgeht).

Mit einem Gleichzeitigkeitsgrad von 1 ist es immer noch möglich, Modelle zu erstellen, die Gerätegruppen als Eingaben verwenden. Diese funktionieren jedoch weiterhin unabhängig für jedes Gerät innerhalb der Gerätegruppe, und es ist noch immer nicht möglich, die Eingaben oder Ausgaben von Gerätegruppen und einzelnen Geräten zu kombinieren.

Anmerkung:

Die Verwendung mehrerer spezifischer Geräte in einem Modell mit einem Gleichzeitigkeitsgrad von mehr als 1 kann zu Verbindungen zwischen Modellen führen, die bei mehreren „Workers“ eingesetzt werden. Modellketten, die mehrere spezifische Geräte mit hohem Durchsatz verwenden, skalieren normalerweise weniger gut als Modellketten, die alle ein einziges spezifisches Gerät verwenden.

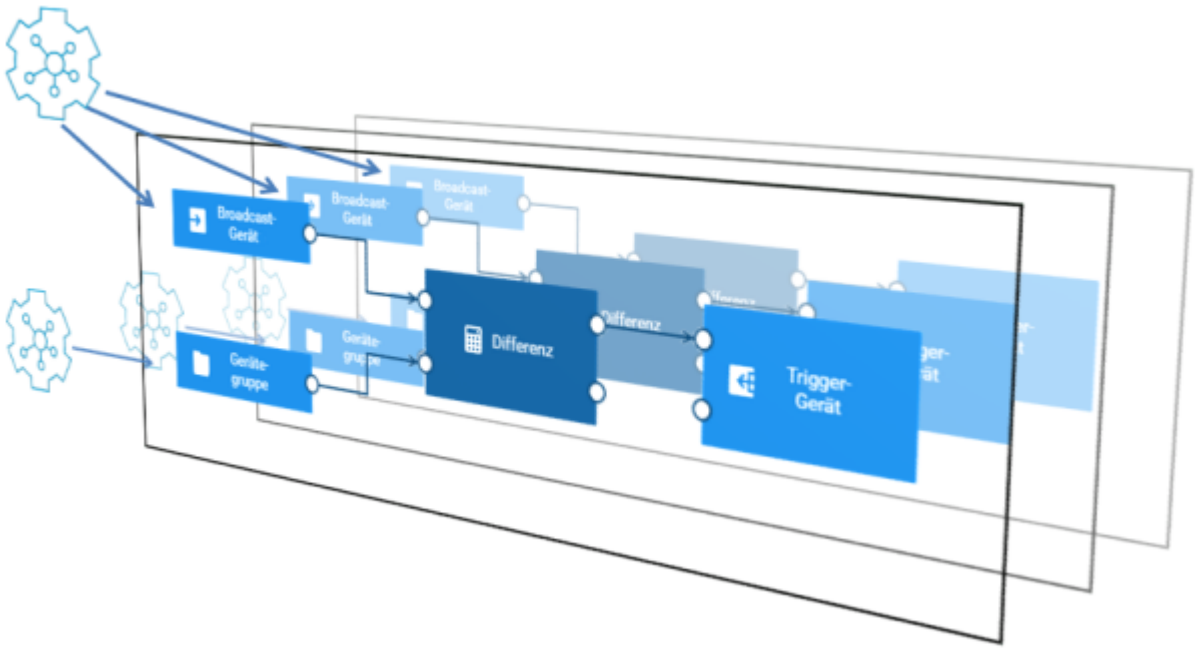
Die Unterstützung mehrerer Geräte in einem einzigen Modell mit einem Gleichzeitigkeitsgrad von mehr als 1 ist deaktiviert, wenn Ein- oder Ausgabeblöcke, die mit der API-Version 1 des Analytics Builder Block SDK geschrieben wurden, als Extensionen hinzugefügt werden. Informationen zur Migration der Blöcke auf die API-Version 2 finden Sie in der Dokumentation unter https://github.com/SoftwareAG/apama-analytics-builder-block-sdk.

Broadcast-Geräte

Es ist manchmal hilfreich, Signale zu haben, die für alle Geräte gelten. Dies können Signale von Geräten sein oder von anderen Systemen, die so präsentiert werden, als wären sie Signale von einem Gerät. Analytics Builder unterstützt daher sogenannte *Broadcast-Geräte*. Die Signale von diesen Geräten stehen allen Modellen und Geräten zur Verfügung.

Broadcast-Geräte können in jedem Modell als Eingaben verwendet werden, entweder zusammen mit den Eingaben für spezifische Geräte oder für Gerätegruppen. Das folgende Diagramm zeigt ein Broadcast-Gerät, das für alle Geräte innerhalb einer Gerätegruppe gilt. Es ist möglich, die Signale von Geräten in einer Gerätegruppe mit den Signalen von einem Broadcast-Gerät zu

kombinieren, indem man sie einem Verarbeitungsblock (wie zum Beispiel dem **Ausdruck**-Block) als unterschiedliche Eingaben übergibt.



Ein Broadcast-Gerät kann - im Gegensatz zu anderen Geräten - nur für synchrone Ausgaben eines Modells verwendet werden, das ausschließlich Daten von Broadcast-Geräten verarbeitet. Broadcast-Ausgaben des asynchronen Typs können von einem Modell generiert werden, das Nicht-Broadcast-Eingaben verarbeitet.

Es ist auch nicht möglich, Modelle mit Hilfe von synchronen Ausgabedaten eines Broadcast-Geräts zu verbinden (das heißt, kein Modell kann einen Messwert von einem Broadcast-Gerät verwenden, das die Ausgabe eines anderen Modells ist). Modelle können über asynchrone Ausgaben eines Broadcast-Geräts miteinander verbunden werden (das heißt, Modelle können eine Operation von einem Broadcast-Gerät verwenden, das die Ausgabe eines anderen Modells ist).

Broadcast-Geräte identifizieren

Broadcast-Geräte werden durch das Vorhandensein einer Eigenschaft auf dem Geräteobjekt im Inventar für dieses Gerät identifiziert. Dies ist die Eigenschaft `pas_broadcastDevice` oder `c8y_Kpi`. Ob ein Gerät als Broadcast-Gerät behandelt wird oder nicht, ist daher über alle Modelle hinweg global. Es ist nicht zulässig, eine Gerätegruppe zu verwenden, die ein Broadcast-Gerät enthält. `c8y_Kpi`-Objekte werden in der Regel mit dem KPI-Block verwendet. So ist es mit einem KPI-Objekt möglich, die Messungen aus einer Gerätegruppe zu vergleichen. Hierbei wird ein KPI-Objekt für alle Geräte in der Gruppe verwendet.

Virtuelle Geräte

Ein virtuelles Gerät wird benutzt, wenn ein Modell im Test- oder Simulationsmodus aktiviert wird. Siehe auch „Ein Modell aktivieren“ auf Seite 56.

Virtuelle Geräte sind Objekte im Cumulocity IoT-Inventar, die die Eigenschaft `c8y_VirtualDevice` haben. Diese Eigenschaft verweist auf die Kennung (ID) des realen Geräts, von dem das virtuelle Gerät die Kopie ist.

Verwenden Sie `creationDate` um herauszufinden, welches `virtualDevice` für eine Modellaktivierung erstellt wurde und bei welchen Messwerten dieses Gerät die Quelle ist.

Standardmäßig werden die virtuellen Geräte 30 Tage lang aufbewahrt. Wenn Sie diese Vorgabe ändern möchten, dann müssen Sie die Mandantenoptionen (englisch „tenant options“) ändern. Das heißt, Sie müssen eine Anfrage mit `POST /tenant/options` senden. Ausführliche Informationen finden Sie bei der Beschreibung der „Tenants“ im *Reference guide* unter https://www.cumulocity.com/guides/. Geben Sie zum Beispiel Folgendes an, wenn Sie die Aufbewahrungsdauer auf einen Tag abändern möchten:

```
{
    "category": "analytics.builder",
    "key": "retention.virtualDevicesMaxDays",
    "value": "1"
}
```

Siehe auch „Konfiguration“ auf Seite 157.

Virtuelle Geräte werden nicht in der Device Management-Anwendung angezeigt. Verwenden Sie REST-Operationen (wie im *Reference guide* unter https://www.cumulocity.com/guides/ beschrieben), um diese Einträge zu finden.

Verbindungen zwischen Modellen

Mit den Aus- und Eingabeblöcken können Sie mehrere Modelle miteinander verbinden. Ein Modell, das einen Ausgabeblock wie den **Messwertausgabe**-Block enthält (für `Measurement`-Objekte von Cumulocity IoT), generiert eine Reihe von Ereignissen, und diese Ereignisse können in einem anderen Modell von einem passenden Eingabeblock (wie zum Beispiel dem **Messwerteingabe**-Block) verarbeitet werden. Siehe auch „Schlüssel zur Identifizierung einer Ereignisreihe“ auf Seite 114.

Wenn Modelle miteinander verbunden sind, die dieselben Ein- und Ausgaben für denselben Ereignisstrom benutzen, dann werden alle diese verbundenen Modelle als „Kette“ bezeichnet. Es kann mehrere Ketten geben, wenn separate Gruppen von Modellen miteinander verbunden sind.

Anmerkung:

Die Ereignisse von einem Modell können nur dann von einem anderen Modell verarbeitet werden, wenn alle beteiligten Modelle im Produktionsmodus aktiv sind. Wenn die Modelle im Test- oder Simulationsmodus aktiv sind, dann werden virtuelle Geräte verwendet, und die Ereignisse von einem Modell können daher nicht von einem anderen Modell verarbeitet werden.

Wenn ein Modell einen synchronen Ausgabeblock hat, der eine Reihe von Ereignissen für einen bestimmten Schlüssel generiert, und wenn ein zweites Modell einen Eingabeblock hat, der dieselbe Ereignisreihe (mit denselben Schlüsselparametern) verarbeitet, dann bildet dies eine Verbindung zwischen dem ersten und dem zweiten Modell. Nachdem das erste Modell seinen Ausgabeblock verarbeitet hat, wertet das zweite Modell die neue Eingabe mit seinem Eingabeblock aus.

Es ist auch möglich, die Verbindungen zwischen den Modellen mit Hilfe der Ausgabe eines asynchronen Ausgabeblocks herzustellen. Wenn in diesem Fall das erste Modell den asynchronen Ausgabeblock auslöst, wird die Ausgabe generiert und an das externe System (zum Beispiel Cumulocity IoT) gesendet. Die Daten von dem externen System werden zu einem späteren Zeitpunkt wieder zurückerhalten und verursachen die Auswertung aller anderen Modelle, die die Daten verwenden.

Ähnlich wie bei der Verarbeitungsreihenfolge der Verbindungen innerhalb eines Modells (siehe auch „Verarbeitungsreihenfolge der Verbindungen“ auf Seite 108), gilt Folgendes, wenn ein Ausgabeblock in einem Modell eine Reihe von Ereignissen generiert, die von einem Eingabeblock in einem anderen Modell verarbeitet werden:

- Ein einzelnes Modell kann dieselben Ereignisse an mehr als ein anderes Modell senden. Somit ist es möglich, dass ein einzelnes Modell eine allgemeine Vorverarbeitung durchführt, wie zum Beispiel die Umrechnung von Einheiten oder die Berechnung eines Durchschnittswerts (mit dem **Durchschnitt (Mittelwert)**-Block) und dass dieser Wert anschließend von mehreren anderen Modellen verwendet wird.
- Die Modelle werden in Reihenfolge ausgeführt, gemäß den Verbindungen zwischen den Modellen, die durch synchrone Ausgaben gebildet wurden. Die Quelle einer Verbindung wird immer vor dem Ziel der Verbindung ausgewertet. Wenn ein Modell Verbindungen von mehreren Blöcken hat, die alle von demselben Ursprungsereignis kommen, dann werden zuerst alle Verbindungen ausgewertet, und das empfangende Modell wertet alle Eingaben anschließend einmal aus.

  Verbindungen, die mit Hilfe von asynchronen Ausgaben hergestellt wurden, haben keine bestimmte Ausführungsreihenfolge. Ein Modell, das die Ausgabe verwendet, wird nur ausgeführt, wenn die Ausgabe vom externen System zurückerhalten wird.

Ähnlich wie bei den Verbindungseinschränkungen in einem Modell (siehe auch „Verbindungseinschränkungen“ auf Seite 109), gibt es Einschränkungen wie die Aus- und Eingangsblöcke zum Verbinden der Modelle benutzt werden können:

- In allen Modellen darf ein einziger Block eine synchrone Ereignisreihe für einen bestimmten Schlüssel generieren. Siehe auch „Schlüssel zur Identifizierung einer Ereignisreihe“ auf Seite 114.Mehrere Ausgabeblöcke, die asynchrone Ereignisse erzeugen, können innerhalb eines einzelnen Modells oder über mehrere Modelle hinweg verwendet werden.
- Bei einer asynchronen Ausgabe können zwischen den Modellen keine Kreisläufe erstellt werden. Ein Modell, das Ereignisse über einen Eingabeblock empfängt, der synchron von einem anderen Modell generiert wurde, kann keinen Ausgabeblock enthalten, der synchrone Ereignisse generiert, die das andere Modell verarbeiten würde. Dies gilt sogar dann, wenn eines der Modelle zwei separate Teile enthält, so dass es keinen tatsächlichen Kreislauf bezüglich der Verbindungen zwischen den Modellen gibt. Kreisläufe zwischen den Modellen können durch asynchrone Ausgaben erstellt werden. Daher müssen Sie darauf achten, dass ihr Modell keine unbestimmten Kreisläufe ausführt.

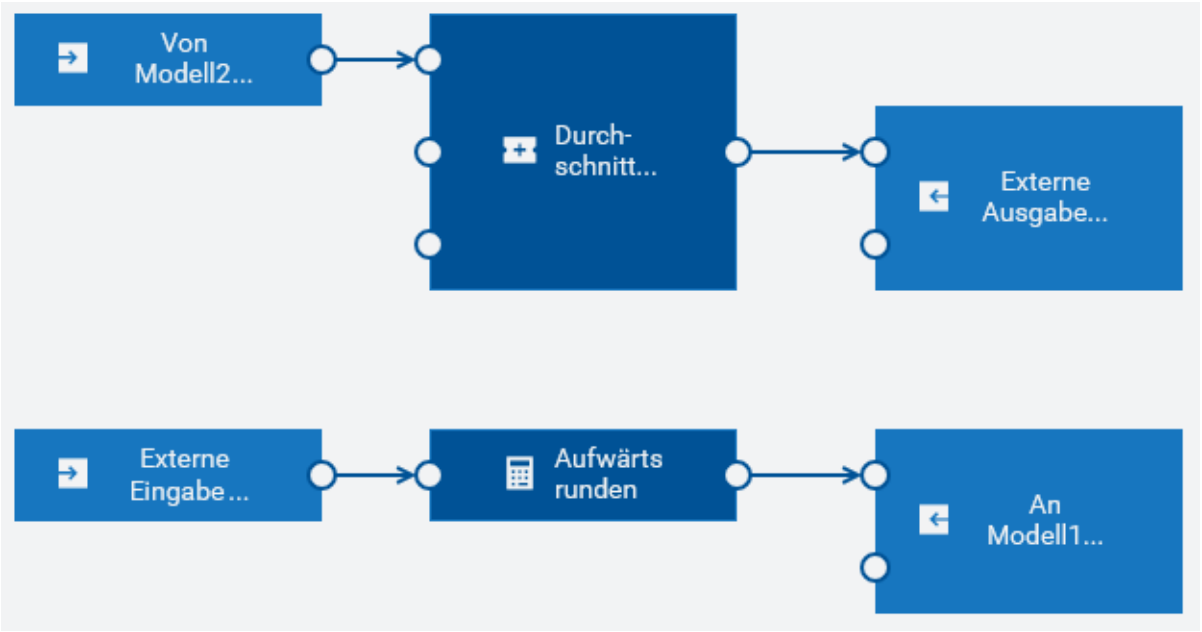
Jedes Modell, das sich nicht an diese Einschränkungen hält, verursacht bei dem Versuch es zu aktivieren einen Fehler, wenn es in Kombination mit den bereits aktivierten Modellen verwendet wird. Dies gilt für das letzte Element in einem Modellkreislauf. Bei einem solchen Fehler liegt das Problem eher in der Interaktion zwischen den Modellen und nicht in einem modellspezifischen Problem; jedoch werden vorhandene, bereits aktivierte Modelle nicht automatisch deaktiviert.

Wenn zum Beispiel mehrere Modelle dieselbe synchrone Ereignisreihe (mit demselben Schlüssel) generieren, dann kann das erste zu aktivierende Modell bereitgestellt werden, aber alle nachfolgenden Modelle melden beim Aktivierungsversuch einen Fehler.

Das folgende Beispiel benutzt drei Modelle: Modell1, Modell2 und Modell3. Ein Kreislauf kann in den folgenden Fällen bestehen:

- Ein Ausgabeblock von Modell1 erzeugt eine Reihe von synchronen Ereignissen, die von einem Eingabeblock in Modell2 verarbeitet werden, und Modell2 enthält einen Ausgabeblock, der eine zweite Reihe von synchronen Ereignissen generiert, und
- Modell3 enthält einen Eingabeblock, der eine Reihe von Ereignissen von Modell2 verarbeitet, und Modell3 enthält auch einen Ausgabeblock, der eine Reihe von synchronen Ereignissen generiert, die von einem Eingabeblock in Modell1 verwendet werden.

In diesem Fall können nur zwei dieser Modelle fehlerfrei aktiviert werden. Bei einer Aktivierung in Reihenfolge hätte nur Modell3 einen Fehler. Wenn Modell1 oder Modell2 jedoch nicht aktiv wären, dann könnte Modell3 aktiviert werden. Der Fehler tritt auch dann auf, wenn eines der Modelle keine Verbindung von dem Eingabeblock in der Kette zu dem Ausgabeblock in derselben Kette hat, wie in dem Beispiel für Modell3 unten: Die Ereignisse von Modell2 bilden bei der Verbindung mit dem **An Modell1 Messung**-Ausgabeblock keinen Kreislauf; sie zählen aber als Kreislauf, weil sich beide im selben Modell befinden. (In diesem Fall könnte das Problem gelöst werden, indem man das Modell in zwei Modelle aufteilt und den Kreislauf somit beseitigt.)



Anmerkung:

Die Verwendung mehrerer spezifischer Geräte in einem Modell mit einem Gleichzeitigkeitsgrad von mehr als 1 kann zu Verbindungen zwischen Modellen führen, die bei mehreren „Workers“ eingesetzt werden. Modellketten, die mehrere spezifische Geräte mit hohem Durchsatz verwenden, skalieren normalerweise weniger gut als Modellketten, die alle ein einziges spezifisches Gerät verwenden.

Die Anzahl der angezeigten Geräte, Gerätegruppen und/oder Assets konfigurieren

Standardmäßig werden in den **Eingabe**- und **Ausgabe**-Kategorien der Palette maximal 10 Geräte und 10 Gerätegruppen angezeigt (siehe auch „Einen Block hinzufügen“ auf Seite 67). Wenn Sie die Suchfelder in diesen Kategorien verwenden, dann gilt diese Standardeinstellung auch für die maximale Anzahl von Assets, die im Suchergebnis angezeigt werden. Wenn Sie auf **Mehr laden** klicken, dann werden bis zu 10 weitere Geräte, Gerätegruppen und/oder Assets angezeigt.

Der gleiche Standardwert wird im Blockparametereditor verwendet, wenn Sie ein anderes Gerät, eine andere Gerätegruppe oder ein anderes Asset auswählen (siehe „Die Parameter eines Blocks editieren“ auf Seite 69) und wenn Sie Geräte, Gerätegruppen oder Assets ersetzen (siehe „Geräte, Gerätegruppen und Assets ersetzen“ auf Seite 76).

Wenn Sie diesen Standardwert ändern möchten (um mehr oder weniger Elemente anzuzeigen), dann müssen Sie die Mandantenoptionen (englisch „tenant options“) ändern. Hierzu müssen Sie eine Anfrage mit `POST /tenant/options` senden. Ausführliche Informationen finden Sie bei der Beschreibung der „Tenants“ im *Reference guide* unter https://www.cumulocity.com/guides/.

Geben Sie beispielsweise Folgendes an, um den Wert auf 20 festzulegen:

```
{
    "category": "analytics.builder",
    "key": "c8yAnalyticsBlocks.queryInventoryPageSize",
    "value": "20"
}
```

Siehe auch „Konfiguration“ auf Seite 157.

Nach Eingabe- und Ausgabe-Assets suchen

Standardmäßig werden in der Palette nur Geräte und Gerätegruppen angezeigt (siehe auch „Einen Block hinzufügen“ auf Seite 67). Wenn Sie jedoch die Suchfelder verwenden, die in der Palette für die Eingabe- und Ausgabeblöcke zur Verfügung stehen, dann werden alle Assets (nicht nur die Geräte und Gerätegruppen) im Cumulocity IoT-Inventar angezeigt, die den Suchkriterien entsprechen. Somit können Sie Analysemodelle erstellen, bei denen jedes beliebige Asset im Inventar als Eingabe- oder Ausgabeblock definiert werden kann.

Wenn Sie die Suche so einschränken möchten, dass nur Elemente eines bestimmten Typs angezeigt werden (zum Beispiel nur Geräte), dann müssen Sie die Mandantenoptionen (englisch „tenant options“) ändern. Hierzu müssen Sie eine Anfrage mit `POST /tenant/options` senden. Ausführliche Informationen finden Sie bei der Beschreibung der „Tenants“ im *Reference guide* unter https://www.cumulocity.com/guides/.

Geben Sie beispielsweise Folgendes an, wenn Sie nur Geräte anzeigen möchten:

```
{
    "category": "analytics.builder",
    "key": "c8yAnalyticsBlocks.queryInventoryNameSearchAdditionalFilter",
    "value": "has(c8y_IsDevice)"
}
```

Das `c8y_IsDevice` im Wert von `value` ist ein sogenanntes *Fragment*. Sie können jedes Fragment angeben, das bei Cumulocity IoT bekannt ist, einschließlich aller Fragmente, die Sie selbst erstellt haben.

Sie können mehrere Werte kombinieren. Wenn Sie zum Beispiel nur Geräte und Gerätegruppen anzeigen möchten, dann können Sie Folgendes angeben:

```
{
    "category": "analytics.builder",
    "key": "c8yAnalyticsBlocks.queryInventoryNameSearchAdditionalFilter",
    "value": "has(c8y_IsDevice) or has(c8y_IsDeviceGroup)"
}
```

Der Standardwert dieser Mandantenoption ist `not has(c8y_IsVirtualDevice)`. Solange Sie diese Mandantenoption nicht ändern, werden virtuelle Geräte nicht angezeigt, da sie in einem Analysemodell keinen Sinn ergeben würden. Wenn Sie den Wert für diese Mandantenoption jedoch ändern, dann müssen Sie alle Assets angeben, die im Suchergebnis angezeigt werden sollen.

Die Mandantenoptionen werden auch im Blockparametereditor verwendet, wenn Sie ein anderes Gerät auswählen (siehe „Die Parameter eines Blocks editieren“ auf Seite 69) und wenn Sie Geräte ersetzen (siehe „Geräte, Gerätegruppen und Assets ersetzen“ auf Seite 76).

Siehe auch „Konfiguration“ auf Seite 157.

10 Modellsimulation

- Allgemeine Informationen zum Simulationsmodus 144
- Simulationsparameter 145
- Die maximale Anzahl von Simulationsmodellen konfigurieren 145
- Eine alternative Datenquelle für die Simulation konfigurieren 145
- Verworfene Eingaben überwachen 147

Allgemeine Informationen zum Simulationsmodus

Sie können ein Modell im Simulationsmodus ausführen, um es mit historischen Eingabedaten laufen zu lassen (zum Beispiel mit Messwerten von Cumulocity IoT). Das ermöglicht es Ihnen, das Verhalten eines neu entwickelten Modells anhand historischer Daten zu testen oder die Feinabstimmung eines vorhandenen Modells vorzunehmen. Oder es ermöglicht es Ihnen, das Modell mit einer Reihe von historischen Daten mit bekannten Eigenschaften zu testen.

Um ein Modell im Simulationsmodus auszuführen, müssen Sie es im Modellmanager aktivieren. Weitere Informationen finden Sie unter „Ein Modell aktivieren“ auf Seite 56.

Anmerkung:

Der Simulationsmodus ist nur für Modelle mit spezifischen Geräten zulässig. Wenn Sie ein Modell mit einer Gerätegruppe simulieren möchten, dann sollten Sie es mit dem Modelleditor so abändern, dass es für ein einzelnes Gerät innerhalb der Gerätegruppe gilt und aktivieren Sie das Modell anschließend im Simulationsmodus.

Wenn ein Modell im Simulationsmodus ausgeführt wird, dann verwendet es Daten von einem virtuellen Gerät (siehe auch „Virtuelle Geräte“ auf Seite 137). Ein simuliertes Modell kann somit neben anderen nicht simulierten Modellen laufen, ohne sie zu beeinträchtigen.

Ein simuliertes Modell wird so ausgeführt, als würde es zum Zeitpunkt der historischen Daten laufen. Die Eingabedaten werden gemäß der Reihenfolge ihrer historischen Zeit verarbeitet. Das simulierte Modell verwendet die historische Zeit auch für die Zeitstempel der generierten Ausgabe.

Ereignisse, Alarme und Operationen werden mit einem Zeitstempel erstellt. Mit der Zeit kann es jedoch zu Aktualisierungen dieser Objekte kommen. Ein Alarm kann zum Beispiel gelöscht oder der Status einer Operation geändert werden. Da der Änderungsverlauf von Ereignissen, Alarmen und Operationen nicht festgehalten wird, wird jedes dieser Objekte nur mit dem ursprünglichen Zeitstempel durchgespielt, mit der neuesten Version seiner Eigenschaften. Änderungen an diesen Objekten werden also nicht durchgespielt und der Simulationsmodus ist nur von eingeschränktem Nutzen, wenn Ihre Modelle von Objektänderungen abhängig sind.

Anmerkung:

Der Simulationsmodus ist nur zulässig für Modelle mit Eingabeblöcken vom Typ **Managed-Object-Eingabe**.

Wenn die Simulation läuft, dann werden historische Daten aus der Cumulocity IoT-Datenbank im Apama-Korrelator durchgespielt. Wenn es zu erheblichen Verzögerungen bei der Datenbankabfrage kommt oder wenn das System stark ausgelastet ist, dann kann dies in Ausnahmefällen dazu führen, dass Eingabedaten verworfen werden. Ein simuliertes Modell verarbeitet die Eingabedaten mit normaler Geschwindigkeit. Wenn die historischen Dateneinträge zum Beispiel eine Sekunde auseinander liegen, dann werden sie im Abstand von einer Sekunde verarbeitet. Dies bedeutet, dass die Simulation eines Modells mit einer Stunde historischer Daten ungefähr eine Stunde Simulationszeit in Anspruch nimmt.

Simulationsparameter

Wenn Sie ein Modell im Simulationsmodus aktivieren möchten, müssen Sie im Modellmanager die Werte für zwei Parameter definieren: die Startzeit und die Endzeit. Diese Werte bestimmen den Zeitraum, aus dem die historischen Daten in das simulierte Modell gesendet werden sollen.

- **Von**

  Startzeit. Der Beginn des Zeitraums aus dem historische Daten in das Modell gesendet werden sollen.

- **Bis**

  Endzeit. Das Ende des Zeitraums aus dem historische Daten in das Modell gesendet werden sollen.

Das Senden von Daten in das simulierte Modell wird beendet, wenn alle historischen Daten aus dem angegebenen Zeitraum gesendet wurden.

Die maximale Anzahl von Simulationsmodellen konfigurieren

Standardmäßig können maximal 3 Simulationsmodelle gleichzeitig aktiviert werden.

Wenn Sie diesen Standardwert ändern möchten (um mehr oder weniger Simulationsmodelle gleichzeitig zu aktivieren), dann müssen Sie die Mandantenoptionen (englisch „tenant options“) ändern. Hierzu müssen Sie eine Anfrage mit `POST /tenant/options` senden. Ausführliche Informationen finden Sie bei der Beschreibung der „Tenants“ im *Reference guide* unter https://www.cumulocity.com/guides/.

Geben Sie beispielsweise Folgendes an, um den Wert auf 5 festzulegen:

```
{
    "category": "analytics.builder",
    "key": "simulation.maxInstances",
    "value": "5"
}
```

Siehe auch „Konfiguration“ auf Seite 157.

Eine alternative Datenquelle für die Simulation konfigurieren

Standardmäßig wird die Plattformdatenbank zum Abrufen der historischen Daten für die Simulation eines Analysemodells verwendet. Sie können eine alternative Datenquelle für die Simulation konfigurieren, zum Beispiel, wenn die historischen Daten separat gehalten werden. Die Datenquelle muss HTTP-`GET`-Operationen für die erforderlichen Pfad- und Abfrageparameter unterstützen. Die Antwort auf die `GET`-Operation muss dem Standard-JSON-Format von Cumulocity IoT entsprechen. Informationen zu den Daten- und Abfrageparameterformaten finden Sie in der Cumulocity IoT- Dokumentation.

Wenn Sie eine alternative Datenquelle für die Simulation verwenden möchten, dann müssen Sie die unten angegebenen Mandantenoptionen (englisch „tenant options“) definieren. Das heißt: Sie müssen 3 separate Anfragen mit `POST /tenant/options` senden. Ausführliche Informationen finden

Sie bei der Beschreibung der „Tenants“ im *Reference guide* unter https://www.cumulocity.com/guides/.

- `simulation.dataSource.url` - Die Basis-HTTP-URL der Datenquelle.
- `simulation.dataSource.username` - Der Benutzername, der für die HTTP-Basisauthentifizierung verwendet werden soll.
- `simulation.dataSource.password` - Das Passwort, das für die HTTP-Basisauthentifizierung verwendet werden soll.

Beispiel:

```
{
    "category": "analytics.builder",
    "key": "simulation.dataSource.url",
    "value": "http://192.168.1.1/"
}
```

```
{
    "category": "analytics.builder",
    "key": "simulation.dataSource.username",
    "value": "myname"
}
```

```
{
    "category": "analytics.builder",
    "key": "simulation.dataSource.password",
    "value": "secret"
}
```

Siehe auch „Konfiguration“ auf Seite 157.

Nachfolgend werden die Pfad- und Abfrageparameter aufgeführt, die von der alternativen Datenquelle unterstützt werden müssen.

Alarme

Pfad: `alarm/alarms`

Abfrageparameter:

- `source` - Source-ManagedObject-ID der Alarme.
- `dateFrom` - Startdatum oder Datum und Uhrzeit des Alarmvorgangs.
- `dateTo` - Enddatum oder Datum und Uhrzeit des Alarmvorgangs.
- `pageSize` - Maximale Anzahl der zurückzugebenden Datensätze.
- `currentPage` - Die aktuell zurückgegebene Seite innerhalb des kompletten Ergebnissets, beginnend ab Seite 1.

Ereignisse

Pfad: `event/events`

Abfrageparameter:

- `source` - Source-ManagedObject-ID der Ereignisse.
- `dateFrom` - Startdatum oder Datum und Uhrzeit der Ereignisse.
- `dateTo` - Enddatum oder Datum und Uhrzeit der Ereignisse.
- `pageSize` - Maximale Anzahl der zurückzugebenden Datensätze.
- `currentPage` - Die aktuell zurückgegebene Seite innerhalb des kompletten Ergebnissets, beginnend ab Seite 1.

Messungen

Pfad: `measurement/measurements`

Abfrageparameter:

- `source` - Source-ManagedObject-ID der Messungen.
- `dateFrom` - Startdatum oder Datum und Uhrzeit der Messungen.
- `dateTo` - Enddatum oder Datum und Uhrzeit der Messungen.
- `pageSize` - Maximale Anzahl der zurückzugebenden Datensätze.
- `currentPage` - Die aktuell zurückgegebene Seite innerhalb des kompletten Ergebnissets, beginnend ab Seite 1.

Operationen

Pfad: `devicecontrol/operations`

Abfrageparameter:

- `deviceId` - Source-Geräte-ID der Operationen.
- `createdFrom` - Starterstellungszeit der Operationen.
- `createdTo` - Enderstellungszeit der Operationen.
- `pageSize` - Maximale Anzahl der zurückzugebenden Datensätze.
- `currentPage` - Die aktuell zurückgegebene Seite innerhalb des kompletten Ergebnissets, beginnend ab Seite 1.

Verworfene Eingaben überwachen

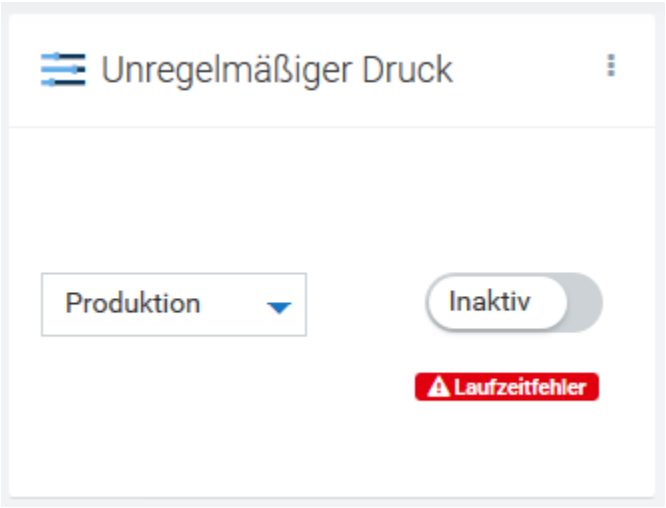
In Ausnahmefällen kann es dazu kommen, dass das simulierte Modell verzögerte Eingabeereignisse verwirft. Die Anzahl der in allen Modellen verworfenen Eingabeereignisse wird als benutzerdefinierter Status mit dem Namen `user-analytics-oldEventsDropped` offengelegt. Siehe auch „Verworfene Ereignisse überwachen“ auf Seite 155.

11 Überwachung und Konfiguration

- Überwachung 150
- Konfiguration 157
- Korrelatorlog anzeigen 161

Überwachung

Im Modellmanager können Sie den aktuellen Status von jedem Modell überwachen. Die Karte eines Modells zeigt den aktuellen Modus (zum Beispiel den Produktionsmodus) und Status (aktiv oder inaktiv) des Modells an.



Wenn bei der Aktivierung oder während der Ausführung eines Modells ein Fehler aufgetreten ist, dann wird Laufzeitfehler auf der Karte dieses Modell angezeigt. Wenn Sie herausfinden möchten, ob der Fehler beim Verarbeiten der Daten aufgetreten ist, dann sollten Sie alle Modelle im Modellmanager erneut laden, damit der jeweils neueste Status auf den Karten angezeigt wird. Siehe auch „Alle Modelle neu laden“ auf Seite 60.

Regelmäßig wiederkehrenden Status überwachen

Zusätzlich zu dem Status, der auf der Karte eines Modell angezeigt wird, können Sie die Generierung eines periodischen Status einschalten, welcher dann über Operationen oder Ereignisse von Cumulocity IoT veröffentlicht wird. Siehe „Konfiguration“ auf Seite 157 zur Einstellung der Mandantenoptionen `status_device_name` und `status_period_secs`.

Jede Operation hat die folgenden Parameter:

Parameter	Beschreibung
`models_running`	Informationen zu den aktiven Modellen, die ausgeführt werden.
`models_failed`	Informationen zu den aktiven Modellen, bei denen ein Fehler aufgetreten ist.
`chain_diagnostics`	Informationen zu den Modellketten. Siehe auch „Verbindungen zwischen Modellen“ auf Seite 138.
`apama_status`	Statusmetriken des Apama-Korrelators. Viele Statusnamen entsprechen den Key-Namen im REST-API von Apama. Die Werte

Parameter	Beschreibung
	werden von der `getValues()`-Aktion des Ereignisses `com.apama.correlator.EngineStatus` ausgegeben und über das REST-API offengelegt.

Modellstatus

Die folgenden Informationen werden für jedes aktive Modell veröffentlicht, das derzeit ausgeführt wird oder bei dem ein Fehler aufgetreten ist:

Name	Beschreibung
`mode`	Der Modus des aktiven Modells. Für Modelle im Simulationsmodus ist dies `SIMULATION`. Ansonsten ist es `PRODUCTION`.
`modeProperties`	Alle modusspezifischen Eigenschaften des Modells. Bei Modellen, die im Modus `SIMULATION` laufen, beinhaltet dies die Start- und Endzeit der Simulation.
`numModelEvaluations`	Die Gesamtanzahl der Male, wie oft das Modell seit der Aktivierung ausgewertet wurde.
`numBlockEvaluations`	Die Gesamtanzahl der Male, wie oft die Blöcke im Modell seit der Aktivierung des Modells ausgewertet wurden. Dies ist die Summe der Anzahl der Auswertungen für jeden Block im Modell.
`avgBlockEvaluations`	Die durchschnittliche Anzahl der Blöcke, die bei jeder Modellauswertung ausgewertet wurden.
`numOutputGenerated`	Die Gesamtanzahl der vom Modell generierten Ausgaben seit der Aktivierung des Modells.

Diese Informationen zu jedem Modell geben einen Einblick in die Leistungsfähigkeit oder den Ablauf der Modelle. Ein Modell mit einer viel größeren Anzahl von `numBlockEvaluations` als ein anderes Modell kann zum Beispiel signalisieren, dass es die meisten Ressourcen verarbeitet, auch wenn es eine geringe Anzahl von `numModelEvaluations` hat. In ähnlicher Weise kann dies dazu genutzt werden, um herauszufinden, ob ein Modell seine Ausgaben mit der erwarteten Rate erzeugt - im Verhältnis dazu wie oft es ausgewertet wird.

Sie können den Status mit dem REST-API oder dem Management Interface (EPL-Plug-in) von Apama überwachen. Weitere Informationen finden Sie in den folgenden Abschnitten der englischsprachigen Produktdokumentation von Apama:

- "Managing and Monitoring over REST" in *Deploying and Managing Apama Applications*, und
- "Using the Management interface" in *Developing Apama Applications*.

Kettendiagnose

Die folgenden Informationen werden für alle vorhandenen Ketten veröffentlicht:

Name	Beschreibung
`creationTime`	Die Zeit, zu der die Kette erstellt wurde.
`executionCount`	Die Anzahl der Male, wie oft die Kette ausgewertet wurde. [1]
`modelsInEvalOrder`	Eine Liste von Modell-IDs, in der Reihenfolge, in der die Modelle ausgewertet wurden.
`pendingTimersCount`	Die Anzahl der ausstehenden Timer, die hinter der aktuellen Zeit liegen.
`maxTime`	Die Höchstzeit, die für die Auswertung der Kette benötigt wurde. [1]
`minTime`	Die Mindestzeit, die für die Auswertung der Kette benötigt wurde. [1]
`meanTime`	Die Durchschnittszeit, die für die Auswertung der Kette benötigt wurde. [1]
`execBucket`	Eine Statistik über die Ausführungszeit der Kette. [1, 2]

- [1] Die Felder werden aktualisiert, wenn die Kette ganz oder teilweise ausgewertet wird. Bei einer teilweisen Auswertung einer Kette werden nur einige Modelle der Kette ausgewertet.
- [2] Es gibt 21 Buckets. In den Buckets wird die Anzahl der Male gespeichert, bei denen die Ausführungszeit innerhalb des Bucket-Bereichs liegt. Jeder Bucket hat eine Größe von `timedelay_secs` geteilt durch 10 Sekunden, mit Ausnahme des letzten Buckets, der sich bis ins Unendliche erstreckt. Beispiel: Wenn `timedelay_secs` 2 Sekunden beträgt, dann enthält der erste Bucket die Anzahl der Male, bei denen die Ausführung der Kette bis zu 0,2 Sekunden dauerte. Der zweite Bucket enthält in diesem Fall die Anzahl der Male, bei denen die Ausführung der Kette mehr als 0,2 Sekunden und bis zu 0,4 Sekunden dauerte. Und so weiter. Siehe auch das folgende Beispiel:

Bucket	Bereich der Ausführungszeit
1	0 - 0,2
2	0,2 - 0,4
3	0,4 - 0,6
...	...
20	3,8 - 4,0

Bucket	Bereich der Ausführungszeit
21	4,0 - unendlich

Weitere Informationen zu `timedelay_secs` finden Sie unter „Keys für Modell-Timeouts“ auf Seite 158.

Status der langsamsten Kette

Wenn Modellketten mit hohem Durchsatz über mehrere Worker hinweg eingesetzt werden, kann es vorkommen, dass eine Kette bei der Verarbeitung der Eingabeereignisse in Rückstand gerät, wodurch ein Rückstandsprotokoll an Eingabeereignissen entsteht, die noch verarbeitet werden müssen. Diese Ketten werden „langsame Ketten“ (englisch: „slow chains“) genannt. Eine Nachricht wird in das Korrelatorlog geschrieben, wenn die langsamste Kette um mehr als 1 Sekunde verzögert ist. Zum Beispiel:

```
Analytics Builder chain of models "Model 1", "Model 2", "Model 3" is slow by 3 seconds.
```

Siehe „Korrelatorlog anzeigen“ auf Seite 161; dort ist beschrieben wo das Korrelatorlog zu finden ist.

Die folgenden Informationen über die langsamste Kette sind auch im periodischen Status verfügbar, der als Cumulocity IoT- Operation oder -Ereignis veröffentlicht wird, im Parameter `apama_status`:

Name	Beschreibung
`user-analyticsbuilder.slowestChain.models`	Alle Modelle, die in der langsamsten Kette enthalten sind.
`user-analyticsbuilder.slowestChain.delaySec`	Die Anzahl der Sekunden, um die Kette bei der Verarbeitung der Eingabeereignisse verzögert ist.

Beispiel

Das folgende Beispiel zeigt die Daten einer Statusoperation, die von Cumulocity IoT veröffentlicht werden:

```
{
    "creationTime": "2021-01-05T21:48:54.620+02:00",
    "deviceId": "6518",
    "deviceName": "apama_status",
    "id": "8579",
    "self": "https://myown.iot.com/devicecontrol/operations/8579",
    "status": "PENDING",
    "models_running": {
        "Package Tracking": {
            "mode": "SIMULATION",
            "modeProperties":{"startTime":1533160604, "endTime":1533160614},
            "numModelEvaluations": 68,
            "numBlockEvaluations": 967,
            "avgBlockEvaluations": 14.2,
            "numOutputGenerated": 50
```

```
        }
    },
    "models_failed": {
        "Build Pipeline ": {
            "mode": "PRODUCTION",
            "numModelEvaluations": 214,
            "numBlockEvaluations": 671,
            "avgBlockEvaluations": 3.13,
            "numOutputGenerated": 4
        }
    },
    "chain_diagnostics": {
        "780858_780858": {
            "creationTime": 1600252455.164188,
            "executionCount": 4,
            "modelsInEvalOrder": ["780858_780858", "780860_780860"],
            "pendingTimersCount": 1,
            "timeData": {
                "execBucket": [2,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
                "maxTime": 0.00014781951904296875,
                "meanTime": 0.0001152356465657552,
                "minTime": 6.29425048828125e-05
            }
        }
    },
    "apama_status": {
       "user-analyticsbuilder.slowestChain.models": "\"Model 1\", \"Model 2\", \"Model
3\"",
        "user-analyticsbuilder.slowestChain.delaySec": "3",
        "user-analytics-oldEventsDropped": "1",
        "numJavaApplications": "1",
        "numMonitors": "27",
        "user-httpServer.eventsTowardsHost": "1646",
        "numFastTracked": "183",
        "user-httpServer.authenticationFailures": "4",
        "numContexts": "5",
        "slowestReceiverQueueSize": "0",
        "numQueuedFastTrack": "0",
        "mostBackedUpInputContext": "<none>",
        "user-httpServer.failedRequests": "4",
        "slowestReceiver": "<none>",
        "numInputQueuedInput": "0",
        "user-httpServer.staticFileRequests": "0",
        "numReceived": "1690",
        "user-httpServer.failedRequests.marginal": "1",
        "numEmits": "1687",
        "numOutEventsUnAcked": "1",
        "user-httpServer.authenticationFailures.marginal": "1",
        "user-httpServer.status": "ONLINE",
        "numProcesses": "48",
        "numEventTypes": "228",
        "virtualMemorySize": "3177968",
        "numQueuedInput": "0",
        "numConsumers": "3",
        "numOutEventsQueued": "1",
        "uptime": "1383561",
        "numListeners": "207",
        "numOutEventsSent": "1686",
        "mostBackedUpICQueueSize": "0",
        "numSnapshots": "0",
```

```
        "mostBackedUpICLatency": "0",
        "numProcessed": "1940",
        "numSubListeners": "207"
    }
}
```

Verworfene Ereignisse überwachen

Wenn ein Modell ein Ereignis empfängt, dann kann es verworfen werden, wenn der Korrelator das Ereignis zu spät liefert oder verarbeitet. Siehe „Eingabeblöcke und Ereignissteuerung“ auf Seite 112.

Die Gesamtzahl der in allen Modellen verworfenen Ereignisse wird regelmäßig als Teil der Statusoperation ausgegeben. Die Anzahl der verworfenen Ereignisse steht als benutzerdefinierter Statuswert mit dem Namen `user-analytics-oldEventsDropped` im Parameter `apama_status` der Statusoperation zur Verfügung. Weitere Informationen finden Sie in „Regelmäßig wiederkehrenden Status überwachen“ auf Seite 150.

Alle verworfenen Eingabeereignisse werden auch an den Kanal `AnalyticsDroppedEvents` gesendet, sodass Sie Ihre eigene Überwachung der verworfenen Ereignisse implementieren können. Ein verworfenes Eingabeelement, das an den Kanal `AnalyticsDroppedEvents` gesendet wurde, wird in ein Ereignis vom Typ `apama.analyticsbuilder.DroppedEvent` gepackt. Auf diese Weise können Sie das ursprüngliche verworfene Ereignis extrahieren und analysieren; so können Sie zum Beispiel die Anzahl der verworfenen Ereignisse pro Gerät kategorisieren. Das kann erreicht werden, indem Sie hierfür EPL schreiben, was auf `DroppedEvent`-Ereignisse hört, nach Geräte-ID und/oder Zeit aggregiert, und die überwachbaren Messergebnisse dann an Cumulocity IoT sendet. Informationen zum Aktivieren (englisch: deploy) von EPL-Anwendungen finden Sie unter http://cumulocity.com/guides/ im *Streaming Analytics guide*.

Den Lebenszyklus des Modell überwachen

Wenn ein Modell erstellt oder entfernt wird oder wenn ein Fehler auftritt, dann wird eine Lebenszyklusmeldung in das Korrelatorlog geschrieben. Die Logeinträge können folgendermaßen aussehen:

```
Model "Build Pipeline" with PRODUCTION mode has started.

Model "Build Pipeline" with PRODUCTION mode has ended.

Model "Build Pipeline" with PRODUCTION mode has failed with an error:
IllegalArgumentException - Error while validating parameters for the
block "toggle" of type "apama.analyticskit.blocks.core.Toggle":
The "Set Delay" must be finite and positive: -1.
```

Die Bereitstellung eines Modells kann bestehende Modelle oder Ketten zu einer neuen Kette kombinieren. Die Bildung einer Kette kann eine Weile dauern, da unter Umständen mehrere bestehende Modelle und Ketten kombiniert werden. Aktivierungsmeldungen werden immer dann in das Korrelatorlog geschrieben, wenn die Aktivierung einer Kette gestartet und abgeschlossen wird. Zum Beispiel:

```
Analytics Builder chain of models "Model 1", "Model 2", "Model 3" is being activated.
Analytics Builder chain of models "Model 1", "Model 2", "Model 3" has been activated.
```

Siehe „Korrelatorlog anzeigen“ auf Seite 161; dort ist beschrieben wo das Korrelatorlog zu finden ist.

Überwachungsprotokolle (Audit-Logs) anzeigen

Modellaktivierungen und -deaktivierungen werden in den Audit-Logs angezeigt. Mit der Administrationsanwendung und dem Audit-API von Cumulocity IoT können Sie auf die Audit-Logs zugreifen. Weitere Informationen zum Zugriff auf die Audit-Logs finden Sie im *User guide* (hiervon gibt es auch eine deutsche Version) und im *Reference guide* unter https://www.cumulocity.com/guides/.

Jeder Audit-Log-Eintrag enthält den Namen des Benutzers, der die Aktion ausgeführt hat, sowie den aktuellen Modus des Modells. Für den Test- und Simulationsmodus werden die Kennung (ID) des virtuellen Geräts und die Moduseigenschaften ausgegeben.

Das folgende Beispiel zeigt einen Audit-Log-Eintrag (mit zusätzlichen Zeilenumbrüchen im `text`-Feld für eine bessere Lesbarkeit):

```
{
    "activity": "Activated model",
    "application": "apama_ctrl",
    "severity": "information",
    "text": "TestingModel1:Activated,
             Mode:SIMULATION,
             OutputDevices:[21628],
             startTime:2019-11-06T09:35:10.000Z,
             endTime:2019-11-06T09:35:43.000Z,
             maxDelaySecsSimulation:2",
    "type": "Inventory",
    "user": "apama_test_cep_admin"
}
```

Diagnoseinformationen anzeigen

Um die Diagnoseinformationen anzuzeigen, benötigen Sie die LESEN-Berechtigung für "Regeln" (englisch: CEP management) in Cumulocity IoT. Einzelheiten zum Verwalten von Berechtigungen finden Sie bei den Informationen zur Administrationsanwendung im *User guide* unter https://www.cumulocity.com/guides/ (hiervon gibt es auch eine deutsche Version).

Anmerkung:
Die ADMIN-Berechtigung für "Regeln" beinhaltet nicht die LESEN-Berechtigung.

Wenn Sie die LESEN-Berechtigung für "Regeln" haben, dann stehen unten im Modellmanager Links zum Herunterladen von Diagnoseinformationen zur Verfügung. Hiermit werden Zip-Dateien heruntergeladen, die den Inhalt der Logdatei, Kopien der EPL-Anwendungen und vieles mehr enthalten.

Die Diagnoseinformationen können bei Problemen oder beim Debuggen von EPL-Anwendungen hilfreich sein. Sie sind auch für den Support hilfreich, wenn Sie einen Support-Fall öffnen.

Ausführliche Informationen zu den verfügbaren Diagnoseinformationen finden Sie im *Streaming Analytics guide* unter https://www.cumulocity.com/guides/. Dort finden Sie auch Informationen zu zusätzlichen Endpunkten für REST-Anfragen.

Konfiguration

Sie können die Einstellungen von Analytics Builder, die sogenannten Mandantenoptionen (englisch „tenant options"), anpassen, indem Sie REST-Anfragen an Cumulocity IoT senden. Die Key-Namen, die Sie mit den REST-Anfragen verwenden können, finden Sie in den nachfolgenden Abschnitten. Zusammen mit dem Key-Namen müssen Sie auch einen Kategorienamen (`category`) angeben; dieser ist immer `analytics.builder`.

Einige konkrete Beispiele finden Sie in „Mandantenoptionen mit curl-Befehlen ändern" auf Seite 160. Sie können jedoch jedes beliebige Tool verwenden.

Um die Mandantenoptionen zu ändern, benötigen Sie die ADMIN-Berechtigung für "Optionen" (englisch: Option management) in Cumulocity IoT. Einzelheiten zum Verwalten von Berechtigungen finden Sie bei den Informationen zur Administrationsanwendung im *User guide* unter https://www.cumulocity.com/guides/ (hiervon gibt es auch eine deutsche Version).

ACHTUNG:

Nachdem Sie eine Mandantenoption mit einer REST-Anfrage geändert haben, wird der Korrelator automatisch neu gestartet. In diesem Fall wird ein Alarm mit dem Schweregrad WICHTIG (englisch: MAJOR) ausgelöst. Sie können den Alarm in der Cockpit-Anwendung auf der Seite **Alarme** anzeigen. Informationen zum Arbeiten mit Alarmen finden Sie im *User guide* (hiervon gibt es auch eine deutsche Version) unter https://www.cumulocity.com/guides/.

Keys für die Statusberichterstattung

Key-Name	Beschreibung
`status_device_name`	Der Name des Cumulocity IoT-Geräts, auf dem die Statusoperationen veröffentlicht werden sollen. Der Standardname ist `apama_status`.
`status_period_secs`	Die Häufigkeit in Sekunden, mit der der Status veröffentlicht werden soll. Der Standardwert ist 0 Sekunden. Dies bedeutet, dass die Statusberichterstattung deaktiviert ist. Sie können die Statusberichterstattung aktivieren, indem Sie die Häufigkeit auf einen positiven Wert einstellen.
`status_send_type`	Wie der Status veröffentlicht werden soll. Der Standardwert ist `OPERATION`. Dies bedeutet, dass der Status als Cumulocity IoT-Operation veröffentlicht wird. Sie können dies in einen der folgenden Werte abändern: ■ `EVENT` - Status als Cumulocity IoT-Ereignis veröffentlichen. ■ `MEASUREMENT` - Status als Cumulocity IoT-Messung veröffentlichen.

Key-Name	Beschreibung
`status_send_keys`	Eine durch Kommas getrennte Liste von Feldnamen, die beim Veröffentlichen des Status verwendet werden sollen. Wenn dies nicht festgelegt oder leer ist, dann wird der Status für alle Felder veröffentlicht. Wenn Sie beispielsweise Folgendes angeben, dann enthält der Status nur die Werte für diese Felder in einer Messung. `numQueuedInput,numListeners,numMonitors`
`status_event_type`	Der Ereignistyp, wenn der Status als Cumulocity IoT-Ereignis veröffentlicht wird, oder der Messtyp, wenn der Status als Cumulocity IoT-Messung veröffentlicht wird. Der Standardtyp ist `apama_status`.
`status_event_text`	Der Ereignistext, wenn der Status als Cumulocity IoT-Ereignis veröffentlicht wird. Der Standardtext ist `Apama Status`.

Keys für Modell-Timeouts

Key-Name	Beschreibung
`timedelay_secs`	Die maximale Verzögerung in Sekunden bevor der Eingabeblock ein Ereignis als alt ansieht. Der Standardwert ist 1 Sekunde. Siehe auch „Eingabeblöcke und Ereignissteuerung“ auf Seite 112.
`logging_throttle_secs`	Die Drosselung der Protokollierung in Sekunden. Regelmäßig wiederkehrende Logmeldungen (zum Beispiel solche, die eine Änderung in der Anzahl der Ereignisse melden, die vom Eingangsblock verworfen wurden) werden nicht häufiger angezeigt als durch diese Konstante definiert. Der Standardwert ist 1 Sekunde. Siehe auch „Eingabeblöcke und Ereignissteuerung“ auf Seite 112.
`minimum_wait_time_secs`	Die Mindestwartezeit in Sekunden. Einige Blöcke können die Ausgabe automatisch generieren, basierend auf der Änderungsrate der Ausgabe. Dies legt eine untere Grenze für die Zeit zwischen solchen Ausgaben fest. Siehe auch „Häufig vorkommende Blockeingänge und Parameter“ auf Seite 110.
`default_timeout_secs`	Das Standardzeitlimit in Sekunden für einfache Anfrageantworten. Dies wird beispielsweise für Anfragen bei Cumulocity IoT verwendet. Der Standardwert ist 10 Sekunden.
`block_promise_timeout_secs`	Das Zeitlimit in Sekunden, um auf die Erledigung von `Promise`-Werten zu warten, die von Blockaktionen wie `$validate` zurückgegeben werden. Ein Block gibt einen `Promise`-Wert von

Key-Name	Beschreibung
	einer Aktion zurück, wenn er eine asynchrone Operation ausführt. Wenn ein `Promise`-Wert nicht innerhalb der konfigurierten Zeit erledigt wird, dann scheitert der Einsatz des Modells. Der Standardwert ist 60 Sekunden.

Keys für den Simulationsmodus

Key-Name	Beschreibung
`simulation.maxInstances`	Die maximale Anzahl von Simulationsmodellen, die gleichzeitig aktiviert werden können. Der Standardwert beträgt 3 Modelle. Siehe auch „Die maximale Anzahl von Simulationsmodellen konfigurieren“ auf Seite 145.

`simulation.dataSource.url`
`simulation.dataSource.username`
`simulation.dataSource.password`
	Die URL einer alternativen Datenquelle für die Simulation. Dies erfordert auch einen Benutzernamen und ein Kennwort für die HTTP-Basisauthentifizierung. Siehe auch „Eine alternative Datenquelle für die Simulation konfigurieren“ auf Seite 145.

Andere Keys

Key-Name	Beschreibung
`numWorkerThreads`	Die Anzahl der Worker-Threads. Der Standardwert ist 1. Siehe auch „Den Gleichzeitigkeitsgrad definieren“ auf Seite 136.
`retention.` `virtualDevicesMaxDays`	Die Aufbewahrungsdauer in Tagen für virtuelle Geräte. Der Standardwert ist 30 Tage. Siehe auch „Virtuelle Geräte“ auf Seite 137.
`c8yAnalyticsBlocks.` `queryInventoryPageSize`	Die Anzahl der Geräte, Gerätegruppen oder Assets, die in der Palette angezeigt werden, sowie im Blockparametereditor, wenn Sie ein anderes Gerät, eine andere Gerätegruppe oder ein anderes Asset auswählen. Der Standardwert ist 10 (das heißt, 10 Geräte und 10 Gerätegruppen, oder 10 Assets). Siehe auch „Die Anzahl der angezeigten Geräte, Gerätegruppen und/oder Assets konfigurieren“ auf Seite 141.

`c8yAnalyticsBlocks.`
`queryInventoryName`
`SearchAdditionalFilter`
	Die Assets, die in der Palette angezeigt werden, wenn Sie das Suchfeld verwenden, sowie im Blockparametereditor. Siehe auch „Nach Eingabe- und Ausgabe-Assets suchen“ auf Seite 141.

Protokollierte Mandantenoptionen

Die Werte einiger Mandantenoptionen werden protokolliert. Diese sind:

- `status_device_name`
- `status_period_secs`
- `timedelay_secs`
- `numWorkerThreads`

Wenn Sie herausfinden möchten, welche Werte derzeit für diese Mandantenoptionen verwendet werden, dann können Sie diese im Korrelatorlog nachsehen. Siehe auch „Korrelatorlog anzeigen“ auf Seite 161.

Mandantenoptionen mit curl-Befehlen ändern

Sie können verschiedene Mandantenoptionen festlegen oder ändern, indem Sie `POST`-Anfragen an Cumulocity IoT senden. Dieser Abschnitt erklärt, wie Sie dies mit dem curl-Befehlszeilentool machen können. Ausführliche Informationen zu curl finden Sie unter https://curl.haxx.se/. Siehe auch die Informationen über „Tenants“ im *Reference guide* unter https://www.cumulocity.com/guides/.

Die Syntax des curl-Befehls hängt von der Umgebung ab, in der Sie arbeiten. Die Syntax für eine Bash-UNIX-Shell ist zum Beispiel wie folgt:

```
curl --user benutzername -X POST -H 'Content-Type: application/json' -d '{"category":
"analytics.builder", "key": "keyname", "value": "wert"}' -k
https://hostname/tenant/options
```

Hierbei gilt Folgendes:

- *benutzername* ist der Name eines Benutzers mit ADMIN-Berechtigung für "Optionen" (englisch: Option management) in Cumulocity IoT. curl fordert Sie zur Eingabe eines Passworts auf. Oder Sie können ein Passwort auch im Argument *benutzername* angeben, indem Sie das Argument um einen Doppelpunkt (:) und das Passwort erweitern. Zum Beispiel:

  ```
  --user Benutzer123:geheimpw
  ```

  Wenn Ihr Mandant keinen eigenen eindeutigen Hostnamen hat, müssen Sie die Mandanten_ID im Argument *benutzername* angeben. Zum Beispiel:

  ```
  --user management/Benutzer123
  ```

  oder

  ```
  --user t12345/Benutzer123
  ```

- *keyname* ist einer der in den vorherigen Abschnitten aufgelisteten Key-Namen.
- *wert* ist der Wert, der für den Key festgelegt werden soll. Dies kann je nach Key eine Zahl oder Zeichenfolge sein.

- *hostname* ist der Hostname Ihres Mandanten, auf dem Ihre Benutzeranwendung bereitgestellt wird.
- `category` ist immer `analytics.builder`.

Zum Beispiel (Bash-Shell):

```
curl --user Benutzer123 -X POST -H 'Content-Type: application/json' -d '{"category":
"analytics.builder", "key": "numWorkerThreads", "value": "4"}' -k
https://mein-mandant/tenant/options
```

Korrelatorlog anzeigen

Der Speicherort des Korrelatorlogs hängt von der Umgebung ab, in der Sie arbeiten:

- Cumulocity IoT Core:

  Mit der Administrationsanwendung von Cumulocity IoT können Sie auf die Logdaten des Korrelators zugreifen. Sie finden diese Daten auf der **Logdaten**-Registerkarte des Apama-ctrl-Microservices. Sie müssen den Microservice abonnieren, damit Sie die Logdaten sehen können. Weitere Informationen zu Microservices und Logdaten finden Sie im *User guide* (hiervon gibt es auch eine deutsche Version) unter https://www.cumulocity.com/guides/.

- Cumulocity IoT Edge:

  Detaillierte Informationen zum Speicherort der Logdatei finden Sie im Handbuch *Cumulocity IoT Edge* unter https://www.cumulocity.com/guides/.

12 Blockreferenz

- Übersicht aller Blöcke 164
- Eingabe 167
- Ausgabe 176
- Logik 184
- Berechnung 186
- Aggregat 203
- Fluss-Manipulation 212
- Werkzeug 221

Übersicht aller Blöcke

Die folgende Tabelle enthält eine kurze Beschreibung aller Blöcke, die aus der Palette des Modelleditors ausgewählt werden können, alphabetisch geordnet (mit Ausnahme der benutzerdefinierten Blöcke, die Sie selbst erstellt haben).

Blockname	Beschreibung
Alarmausgabe	Erstellt ein neues `Alarm`-Objekt für ein angegebenes Gerät oder für das aktuelle Gerät mit einem vorkonfigurierten Alarmnamen und Parametern.
Alarmeingabe	Empfängt `Alarm`-Objekte von einem Gerät oder Gerätegruppe und sortiert diese nach dem Zeitstempel.
Ausdruck	Bewertet einen Ausdruck, um arithmetische oder logische Berechnungen oder Zeichenfolgen-Operationen durchzuführen.
Bereich	Vergleicht den Eingabewert mit den definierten unteren und oberen Bereichswerten, um zu erkennen, ob die Eingabe innerhalb oder außerhalb des Bereichs liegt oder ob sie den Bereich überschreitet.
Bereichssuche	Findet den Bereich, in dem der Eingabewert liegt.
Cron Timer	Sendet eine Signalausgabe basierend auf einer Cron-ähnlichen periodischen Timer-Syntax.
Dauer	Misst die Zeit, die seit einer festgelegten Startzeit vergangen ist.
Delta	Berechnet die Differenz zwischen aufeinander folgenden Eingabewerten.
Differenz	Berechnet die absolute und signierte Differenz zwischen den verbundenen Eingaben.
Diskrete Statistik	Erzeugt Statistiken der Summe, Anzahl, Durchschnitt (Mittelwert), Standardabweichung, Minimum und Maximum für diskrete Eingabewerte.
Durchschnitt (Mittelwert)	Berechnet den Mittelwert der Werte im Laufe der Zeit.
E-Mail senden	Sendet eine E-Mail an die angegebenen E-Mail-Adressen.
Eigenschaften festlegen	Gibt einen Pulse mit Eigenschaften aus, die aus Werten an den Eingangsports festgelegt wurden.
Eigenschaft extrahieren	Extrahiert die angegebene Eigenschaft aus dem Eingabewert und konvertiert sie in den angegebenen Typ.
Ereignisausgabe	Erstellt ein neues `Event`-Objekt für das verbundene Gerät oder das Trigger-Gerät.
Ereigniseingabe	Empfängt `Event`-Objekte von einem Gerät oder einer Gerätegruppe und sortiert diese nach dem Zeitstempel.

Blockname	Beschreibung
Fehlende Daten	Erzeugt eine Ausgabe, wenn innerhalb einer bestimmten Zeit keine Eingabe erfolgt ist.
Gate	Hindert die Eingabe daran, zur Ausgabe zu werden, sofern das Gate nicht geöffnet und eingeschaltet ist.
Geofence	Vergleicht den Eingabewert mit dem definierten Geofence-Wert, um zu erkennen, ob sich das Gerät innerhalb des Geofence befindet, und ob das Gerät den Geofence betreten oder verlassen hat.
Gradient	Berechnet den gewichteten linearen Regressionsgradienten für die Werte.
Gruppenstatistik	Erzeugt regelmäßig wiederkehrende Aggregatwerte für alle Geräte in einer Gruppe, für die der Block Eingabewerte erhalten hat.
Integral	Berechnet den Integralwert des Eingabewertes im Laufe der Zeit.
Kombinierer	Berechnet die Ausgabe auf Grundlage des gewählten Modus und der verbundenen Eingänge.
Konstanter Wert	Gibt einen Wert aus, entweder wenn der Eingangsport "Trigger" ein Signal empfängt oder beim Start.
KPI	Vergleicht einen Wert entweder mit einem KPI (Key Performance Indicator) oder dem Datenpunkt eines Geräts.
Latch-Werte	Rastet den aktuellen Eingabewert ein, der empfangen wurde, während der Block eingeschaltet ist.
Limit	Gibt einen Wert aus, der innerhalb des definierten unteren und oberen Limits bleibt.
Machine Learning	Ruft das angegebene Machine Learning-Modell auf, das die Eingabedaten bewertet.
Managed-Object-Ausgabe	Aktualisiert ein `ManagedObject`-Objekt für ein festgelegtes Gerät oder für das aktuelle Gerät.
Managed-Object-Eingabe	Empfängt `ManagedObject`-Objekte von einem Gerät oder Gerätegruppe.
Messwertausgabe	Erstellt ein neues `Measurement`-Objekt für das verbundene Gerät oder das Trigger-Gerät.
Messwerteingabe	Empfängt `Measurement`-Objekte von einem Gerät oder einer Gerätegruppe und sortiert diese nach dem Zeitstempel.
Minimum / Maximum	Berechnet Minimum und Maximum eines Wertes im Laufe der Zeit.
Nach Basis N	Wandelt einen Float in einen Basis N-String um.
NICHT	Führt ein logisches 'NICHT' für die Eingabe aus.

Blockname	Beschreibung
ODER	Führt ein logisches 'ODER' für die Eingaben aus.
Operationsausgabe	Erstellt ein neues `Operation`-Objekt für ein angegebenes oder für das aktuelle Gerät.
Operationseingabe	Empfängt `Operation`-Objekte von einem Gerät oder Gerätegruppe.
Positionseingabe	Empfängt `Event`-Objekte aus einem Gerät oder einer Gerätegruppe und extrahiert das `c8y_Position`-Fragment in ein `Value`-Objekt.
Pulse	Wandelt eine Nicht-Pulse-Eingabe in eine Pulse-Ausgabe um.
Richtungserkennung	Erkennt, ob der Eingabewert die Richtung ändert.
Rundung	Rundet den Eingabewert unter Verwendung einer wählbaren Regel auf eine bestimmte Anzahl an Dezimalstellen oder auf eine Ganzzahl.
Schalter	Gibt die Werte aus einem gegebenen Eingang aus oder dient als Schutzschalter.
Schwelle	Vergleicht den Eingabewert mit dem definierten Schwellenwert, um zu erkennen, ob der Eingabewert den Schwellenwert durchbricht oder überschreitet.
Selektor	Gibt einen Parameterwert aus, abhängig davon, welcher Eingangsport einen `true`-Wert hat, wobei die niedrigste Zahl Vorrang hat.
SMS senden	Sendet eine SMS (Short Message Service) an die angegebene Telefonnummer.
Standardabweichung	Berechnet die Standardabweichung und die Varianz der Werte im Laufe der Zeit.
Textersetzung	Ersetzt Identifizierer, die mit einem Hash und geschweiften Klammern gekennzeichnet sind (zum Beispiel, `#{name}`) in der Textvorlage durch entsprechende Eingaben aus den Eingabewerten.
Überschreitungszähler	Erkennt und zählt die Anzahl der Schwellenüberschreitungen in der angegebenen Richtung.
Umschalten	Konvertiert zwei Pulse-Eingaben in eine Boolean-Ausgabe auf Grundlage der Pulse-zum Setzen und Zurücksetzen. Verzögerungen können optional eingerichtet werden.
UND	Führt ein logisches 'UND' für die Eingaben aus.
Von Basis N	Wandelt einen Basis N-String in einen Float um.
Zähler	Zählt die Gesamteingaben und die wiederholten Eingaben.
Zeitverzögerung	Verzögert die Eingabe um die angegebene Zeit.

Eingabe

Alarmeingabe

`apama.analyticskit.blocks.cumulocity.AlarmInput`

Empfängt `Alarm`-Objekte von einem Gerät oder Gerätegruppe und sortiert diese nach dem Zeitstempel.

Ist der Alarmstatus "Aktiv", werden die Alarme auf Grundlage des Zeitstempels neu angeordnet (und verworfen wenn veraltet), falls "Zeitstempel ignorieren" nicht aktiviert wurde. Andernfalls werden die Daten verarbeitet, sobald sie empfangen werden.

Die Parameter, die den Eingangs-Stream des Blocks definieren, sind "Gerät oder Gerätegruppe" und "Alarmtyp". Wenn dieser Block mit denselben Parametern "Gerät oder Gerätegruppe" und "Alarmtyp" konfiguriert wird wie ein Block "Alarmausgabe" in einem anderen Modell, dann wird eine Verbindung zwischen den Modellen hergestellt, da jeder Block auf denselben Stream von `Alarm`-Objekten verweist.

Hinweis: Bei der Ausführung im Simulationsmodus, muss der Alarmstatus "Aktiv" sein, da nur die Erstellungszeit des Alarms gespeichert wird.

Parameter

Name	Beschreibung	Typ	Hinweise
Alarmtyp	Der Alarmtyp, auf den der Block hört.	string	
Gerät oder Gerätegruppe	Das Gerät oder die Gerätegruppe, von dem/der der Alarm empfangen wurde. Der Modelleditor verwendet den aktuellen Geräte- oder Gerätegruppennamen. Dieser ist intern der Inventar-ID zugeordnet.	string	
Schweregrad	Der Schweregrad des Alarms. Wenn nicht festgelegt, hört der Block auf alle Alarm-Schweregrade.	Option - eine von: ■ Kritisch ■ Wichtig ■ Weniger wichtig ■ Warnung	Optional
Alarmstatus	Der Status des Alarms. Wenn kein Status festgelegt ist, hört der Block auf Alarme mit jedem Status.	Option - eine von: ■ Aktiv	Optional

Name	Beschreibung	Typ	Hinweise
		■ Bestätigt ■ Aufgehoben	
Benachrichtigungs-modus	Filtert `Alarm`-Ereignisse so, dass nur neue Alarme, aktualisierte Alarme oder alle Alarme verarbeitet werden. In der Standardeinstellung werden alle Alarme verarbeitet.	Option - eine von: ■ Alle ■ Nur aktualisierte Alarme ■ Nur neue Alarme	Standard: Alle
Zeitstempel ignorieren	Wenn gewählt, wird der Zeitstempel des eingehenden Alarms ignoriert. Hinweis: Bei der Ausführung im Simulationsmodus werden die Zeitstempel aufgrund der Verwendung historischer Eingabedaten nicht ignoriert.	boolean	Standard: false

Details zum Ausgangsport

Name	Beschreibung	Typ
Alarme	Erzeugt eine Pulse-Ausgabe für jedes empfangene `Alarm`-Objekt, jeweils mit zusätzlichen Eigenschaften.	pulse

Ereigniseingabe

`apama.analyticskit.blocks.cumulocity.DeviceEventInput`

Empfängt `Event`-Objekte von einem Gerät oder einer Gerätegruppe und sortiert diese nach dem Zeitstempel.

Wenn der Parameter "Zeitstempel ignorieren" festgelegt ist, ignoriert der Block den Zeitstempel des Ereignisses und verarbeitet die Ereignisse in der Reihenfolge ihres Empfangs; andernfalls werden alte Ereignisse verworfen.

Die Parameter, die den Eingangs-Stream des Blocks definieren, sind "Gerät oder Gerätegruppe" und "Ereignistyp". Wenn dieser Block mit denselben Parametern "Gerät oder Gerätegruppe" und "Ereignistyp" konfiguriert wird wie ein Block "Ereignisausgabe" in einem anderen Modell, dann wird eine Verbindung zwischen den Modellen hergestellt, da jeder Block auf denselben Stream von `Event`-Objekten verweist.

Hinweis: Bei der Ausführung im Simulationsmodus werden die Zeitstempel aufgrund der Verwendung historischer Eingabedaten nicht ignoriert.

Hinweis: Für `Event`-Objekte wird keine Änderungshistorie gepflegt, sodass es nicht möglich ist, die Originalobjekte aus dem Inventar abzurufen. Deshalb kann sich ein Modell, das diesen Eingabeblocktyp enthält, im Simulationsmodus anders verhalten als im Produktionsmodus.

Parameter

Name	Beschreibung	Typ	Hinweise
Gerät oder Gerätegruppe	Das Gerät oder die Gerätegruppe, von dem/der das Ereignis empfangen wurde. Der Modelleditor verwendet den aktuellen Geräte- oder Gerätegruppennamen. Dieser ist intern der Inventar-ID zugeordnet.	string	
Ereignistyp	Der Ereignistyp, auf den der Block hört.	string	
Benachrichtigungsmodus	Filtert `Event`-Ereignisse so, dass nur neue Ereignisse, aktualisierte Ereignisse oder alle Ereignisse verarbeitet werden. In der Standardeinstellung werden alle Ereignisse verarbeitet.	Option - eine von: ■ Alle ■ Nur aktualisierte Alarme ■ Nur neue Ereignisse	Standard: Alle
Zeitstempel ignorieren	Wenn gewählt, wird der Zeitstempel des eingehenden Ereignisses ignoriert. Hinweis: Bei der Ausführung im Simulationsmodus werden die Zeitstempel aufgrund der Verwendung historischer Eingabedaten nicht ignoriert.	boolean	Standard: false

Details zum Ausgangsport

Name	Beschreibung	Typ
Ereignisse	Erzeugt eine Signalausgabe für jedes empfangene `Event`-Objekt mit zusätzlichen Eigenschaften.	pulse

Managed-Object-Eingabe

`apama.analyticskit.blocks.cumulocity.ManagedObjectInput`

Empfängt `ManagedObject`-Objekte von einem Gerät oder Gerätegruppe.

Der Block führt keine Neusortierung der empfangenen `ManagedObject`-Objekte durch und verarbeitet sie in der Reihenfolge ihres Empfangs. Wenn der Parameter "Eigenschaftsname" angegeben ist, erzeugt der Block keine neue Ausgabe, wenn sich der Wert der angegebenen Eigenschaft seit der letzten Ausgabe nicht geändert hat, selbst wenn sich andere Eigenschaften bei demselben `ManagedObject`-Objekt geändert haben.

Die Wert-Ausgabe aus dem Block enthält alle Eigenschaften zum `ManagedObject`-Objekt, einschließlich der Eigenschaft, die durch den Parameter "Eigenschaftsname" festgelegt wurde. Eigenschaftswerte können über den Block "Eigenschaft extrahieren" aufgerufen werden.

Eigenschaften des Typs `string`, `boolean` oder `float` können durch Angabe des Namens der Eigenschaft im Block "Eigenschaft extrahieren" aufgerufen werden. Beispiel: Wenn der Name der Eigenschaft `ap_State` lautet, geben Sie `ap_State` als Wert für den Parameter "Eigenschaftspfad" des Blocks "Eigenschaft extrahieren" an.

Wenn ein Eigenschaftswert vom Typ JSON-Objekt oder -Sequenz vorliegt, können verschachtelte Werte durch Angabe des vollständigen Pfads zu den verschachtelten Werten als Name der Eigenschaft aufgerufen werden.

Beispiel: Wenn der Name der Eigenschaft `c8y_SpeedMeasurement` lautet und der Wert `{ "Speed": { "value": 1234, "unit": "km/h" } }` (in JSON-Form) vorliegt, geben Sie `c8y_SpeedMeasurement.Speed.unit` als Wert für den Parameter "Eigenschaftspfad" des Blocks "Eigenschaft extrahieren" an, um den Wert der Einheit zu extrahieren.

Alle mit dem `ManagedObject`-Objekt verbundenen Positionsdaten sind als `c8y_Position`-Eigenschaft verfügbar und können über den Block "Eigenschaft extrahieren" extrahiert werden.

Wenn der Wert der Eigenschaft, die durch den Parameter "Eigenschaftsname" festgelegt ist, vom Typ `string`, `boolean` oder `float` ist, so ist der Wert auch direkt im Ausgangsport "Wert" verfügbar und kann direkt von Blöcken verbraucht werden, die Werte dieses Typs verbrauchen, ohne den Block "Eigenschaft extrahieren" zu verwenden, beispielsweise die Blöcke "Ausdruck" oder "Differenz".

Die Parameter, die den Eingangs-Stream des Blocks definieren, sind "Gerät oder Gerätegruppe" und "Eigenschaftsname".

Parameter

Name	Beschreibung	Typ	Hinweise
Gerät oder Gerätegruppe	Das Gerät oder die Gerätegruppe, von dem/der das Managed Object empfangen wurde. Der Modelleditor verwendet den aktuellen Geräte- oder Gerätegruppennamen. Dieser ist intern der Inventar-ID zugeordnet.	string	
Eigenschafts-name	Der Name der Eigenschaft, auf die gehört werden soll.	string	Optional

Name	Beschreibung	Typ	Hinweise
	Das `ManagedObject`-Objekt muss eine Eigenschaft mit diesem Namen haben, andernfalls wird es ignoriert. Wenn nicht definiert, werden die Objekte nicht gefiltert - jede Aktualisierung erzeugt eine Pulse-Ausgabe mit allen Eigenschaften von `ManagedObject`.		
Startwert erfassen	Gibt den Anfangswert aus, wenn das Modell aktiviert ist.	boolean	Standard: false

Details zum Ausgangsport

Name	Beschreibung	Typ
Wert	Generiert für jedes empfangene `ManagedObject`-Objekt eine Ausgabe. Alle Eigenschaften des Managed Object sind als zusätzliche Eigenschaften verfügbar. Auf ihre Werte können Sie über den Block "Eigenschaft extrahieren" zugreifen.	any

Messwerteingabe

`apama.analyticskit.blocks.cumulocity.DeviceMeasurementInput`

Empfängt `Measurement`-Objekte von einem Gerät oder einer Gerätegruppe und sortiert diese nach dem Zeitstempel.

Wenn der Parameter "Zeitstempel ignorieren" festgelegt ist, ignoriert der Block den Zeitstempel des Messwerts und verarbeitet die Messwerte in der Reihenfolge ihres Empfangs; andernfalls werden alte Messwerte verworfen.

Falls eine Gruppe für die Eingabe verwendet wird, wählen Sie ein Gerät innerhalb der Gruppe aus, um Fragment und Serie auszuwählen und wechseln dann zu der gewünschten Gruppe.

Die Parameter, die den Ausgabe-Stream des Blocks definieren, sind "Gerät oder Gerätegruppe" und "Fragment und Serie". Wenn dieser Block mit denselben Parametern "Gerät oder Gerätegruppe" und "Fragment und Serie" konfiguriert wird wie ein Block "Messwertausgabe" in einem anderen Modell, dann wird eine Verbindung zwischen den Modellen hergestellt, da jeder Block auf denselben Stream von `Messwert`-Objekten verweist.

Hinweis: Bei der Ausführung im Simulationsmodus werden die Zeitstempel aufgrund der Verwendung historischer Eingabedaten nicht ignoriert.

Parameter

Name	Beschreibung	Typ	Hinweise
Gerät oder Gerätegruppe	Das Gerät oder die Gerätegruppe, von dem/der der Messwert empfangen wurde. Der Modelleditor verwendet den aktuellen Geräte- oder Gerätegruppennamen. Dieser ist intern der Inventar-ID zugeordnet.	string	
Fragment und Serie	Das Fragment, auf das der Block hört. Zeigt nur Fragmente und Serien für Messwerte, die mit dem ausgewählten Objekt (Gerät oder Gruppe) verbunden sind. Messungen von einem Gerät innerhalb einer Gruppe werden nur angezeigt, wenn ein Gerät ausgewählt wurde (soweit es keine Messungen gibt, bei denen die Gruppe als Quelle dient).	string	
Zeitstempel ignorieren	Wenn gewählt, wird der Zeitstempel des eingehenden Messwerts ignoriert.Hinweis: Bei der Ausführung im Simulationsmodus werden die Zeitstempel aufgrund der Verwendung historischer Eingabedaten nicht ignoriert.	boolean	Standard: false

Details zum Ausgangsport

Name	Beschreibung	Typ
Wert	Der numerische Wert des Measurement-Objekts.	float

Operationseingabe

`apama.analyticskit.blocks.cumulocity.OperationInput`

Empfängt `Operation`-Objekte von einem Gerät oder Gerätegruppe.

Der Block führt keine Neusortierung der empfangenen `Operation`-Objekte durch und verarbeitet Operationen in der Reihenfolge ihres Empfangs. Optional kann der Block so konfiguriert werden, dass er nur Operationen ausführt, die einen bestimmten Status oder eine bestimmte Eigenschaft haben.

Die Ausgabe aus dem Block enthält alle Eigenschaften zum `Operation`-Objekt. Eigenschaftswerte können über den Block "Eigenschaft extrahieren" aufgerufen werden.

Eigenschaften des Typs `string`, `boolean` oder `float` können durch Angabe des Namens der Eigenschaft im Block "Eigenschaft extrahieren" aufgerufen werden. Beispiel: Wenn der Name der

Eigenschaft `ap_State` lautet, geben Sie `ap_State` als Wert für den Parameter "Eigenschaftspfad" des Blocks "Eigenschaft extrahieren" an.

Wenn ein Eigenschaftswert vom Typ JSON-Objekt oder -Sequenz vorliegt, können verschachtelte Werte durch Angabe des vollständigen Pfads zu den verschachtelten Werten als Name der Eigenschaft aufgerufen werden.

Beispiel: Wenn der Name der Eigenschaft `c8y_SpeedMeasurement` lautet und der Wert `{ "Speed": { "value": 1234, "unit": "km/h" } }` (in JSON-Form) vorliegt, geben Sie `c8y_SpeedMeasurement.Speed.unit` als Wert für den Parameter "Eigenschaftspfad" des Blocks "Eigenschaft extrahieren" an, um den Wert der Einheit zu extrahieren.

Der Parameter, der den Eingangs-Stream des Blocks definiert, ist "Gerät oder Gerätegruppe".

Hinweis: Für `Operation`-Objekte wird keine Änderungshistorie gepflegt, sodass es nicht möglich ist, die Originalobjekte aus dem Inventar abzurufen. Deshalb kann sich ein Modell, das diesen Eingabeblocktyp enthält, im Simulationsmodus anders verhalten als im Produktionsmodus.

Parameter

Name	Beschreibung	Typ	Hinweise
Gerät oder Gerätegruppe	Das Gerät oder die Gerätegruppe, von dem/der die Operation empfangen wurde. Der Modelleditor verwendet den aktuellen Geräte- oder Gerätegruppennamen. Dieser ist intern der Inventar-ID zugeordnet.	string	
Name der Operation	Der Name der Operation, auf die der Block hören soll. Wenn festgelegt, muss das `Operation`-Objekt eine Eigenschaft mit diesem Namen haben, andernfalls wird es ignoriert.	string	Optional
Operationsstatus	Der Status, auf den gehört werden soll. Wenn nicht festgelegt, hört der Block auf Operationen mit jedem Status.	Option - eine von: ■ ERFOLGREICH ■ ABGEBROCHEN ■ WIRD AUSGEFÜHRT ■ AUSSTEHEND	Optional
Benachrichtigungs-modus	Filtert `Operation`-Ereignisse so, dass nur neue Operationen, aktualisierte Operationen oder alle Operationen verarbeitet werden. In der	Option - eine von: ■ Alle	Standard: Alle

Name	Beschreibung	Typ	Hinweise
	Standardeinstellung werden alle Operationen verarbeitet.	■ Nur aktualisierte Alarme ■ Nur neue Operationen	

Details zum Ausgangsport

Name	Beschreibung	Typ
Operationen	Generiert für jedes empfangene `Operation`-Objekt eine Pulse-Ausgabe. Alle Eigenschaften des Objekts `Operation` sind als zusätzliche Eigenschaften verfügbar. Auf ihre Werte können Sie über den Block "Eigenschaft extrahieren" zugreifen.	pulse

Positionseingabe

`apama.analyticsbuilder.blocks.PositionInput`

Empfängt `Event`-Objekte aus einem Gerät oder einer Gerätegruppe und extrahiert das `c8y_Position`-Fragment in ein `Value`-Objekt.

Wenn kein `c8y_Position`-Fragment vorhanden ist, wird das Ereignis ignoriert. Wenn das Fragment nicht mindestens einen gültigen Breitengrad und einen gültigen Längengrad enthält, wird das Ereignis ignoriert. Wenn der Parameter "Primärwert" auf "Höhe" gesetzt ist und das Fragment keine Höhe enthält, wird das Ereignis ignoriert. Breitengrade müssen zwischen -90 und 90 Grad einschließlich liegen. Längengrade müssen zwischen -180 und 180 Grad einschließlich liegen.

Der Primärwert der Ausgabe des `Value`-Objekts kann als Breitengrad, Längengrad oder Höhe festgelegt werden. Alle Bestandteile des `c8y_Position`-Fragments werden zum Eigenschaften-Dictionary des `Value`-Objekts hinzugefügt.

Wenn der Parameter "Zeitstempel ignorieren" festgelegt ist, ignoriert der Block den Zeitstempel des Ereignisses und verarbeitet die Messwerte in der Reihenfolge ihres Empfangs. Falls nicht, ordnet er die Ereignisse neu und verwirft alte Messwerte.

Der Parameter, der den Eingangs-Stream des Blocks definiert, ist "Gerät oder Gerätegruppe" und "Ereignistyp", wenn festgelegt.

Hinweis: Bei der Ausführung im Simulationsmodus werden die Zeitstempel aufgrund der Verwendung historischer Eingabedaten nicht ignoriert.

Hinweis: Für `Event`-Objekte wird keine Änderungshistorie gepflegt, sodass es nicht möglich ist, die Originalobjekte aus dem Inventar abzurufen. Deshalb kann sich ein Modell, das diesen Eingabeblocktyp enthält, im Simulationsmodus anders verhalten als im Produktionsmodus.

Parameter

Name	Beschreibung	Typ	Hinweise
Gerät oder Gerätegruppe	Das Gerät oder die Gerätegruppe, von der die Position empfangen wird. Der Modelleditor verwendet den aktuellen Geräte- oder Gerätegruppennamen. Dieser ist intern der Inventar-ID zugeordnet.	string	
Ereignistyp	Der Ereignistyp, auf den der Block hört. Wenn nicht gesetzt, erfolgt keine Filterung nach Typ. Um Ereignisse aus einem anderen Modell zu verwenden, muss diese Eigenschaft festgelegt sein.	string	Optional
Benachrichtigungsmodus	Filtert `Event`-Ereignisse so, dass nur neue Ereignisse, aktualisierte Ereignisse oder alle Ereignisse verarbeitet werden. In der Standardeinstellung werden alle Ereignisse verarbeitet.	Option - eine von: ■ Alle ■ Nur aktualisierte Alarme ■ Nur neue Ereignisse	Standard: Nur neue Ereignisse
Zeitstempel ignorieren	Wenn gewählt, wird der Zeitstempel des eingehenden Messwerts ignoriert.Hinweis: Bei der Ausführung im Simulationsmodus werden die Zeitstempel aufgrund der Verwendung historischer Eingabedaten nicht ignoriert.	boolean	Standard: false
Primärwert	Der Primärwert, der vom Block ausgegeben wird: Breitengrad, Längengrad oder Höhe, oder leer, wenn nicht festgelegt.	Option - eine von: ■ Breitengrad ■ Längengrad ■ Höhe	Optional

Details zum Ausgangsport

Name	Beschreibung	Typ
Position	Ein Objekt, das mindestens Breitengrad und Längengrad enthält.	any

Ausgabe

Alarmausgabe

`apama.analyticskit.blocks.cumulocity.CreateAlarm`

Erstellt ein neues `Alarm`-Objekt für ein angegebenes Gerät oder für das aktuelle Gerät mit einem vorkonfigurierten Alarmnamen und Parametern.

Dieser Block erzeugt eine synchrone Ausgabe. Die Parameter, die den Ausgabe-Stream definieren, sind "Gerät oder Trigger-Gerät" und "Alarmtyp". Die Verwendung einer anderen Instanz dieses Blocks mit demselben Ausgabe-Stream ist nicht erlaubt.

Wenn "CreateAlarm" wiederholt angegeben wird, dann wird der Zähler des Alarms erhöht (sofern er nicht aufgehoben wurde). Wenn der Schweregrad sich geändert hat, wird die Anzahl nicht erhöht - stattdessen wird der vorhandene Alarm geändert.

Parameter

Name	Beschreibung	Typ	Hinweise
Gerät oder Trigger-Gerät	Das Gerät (oder bei Modellen, die mit Gruppen zu tun haben, das Trigger-Gerät), mit dem der Alarm verbunden ist. Der Modelleditor verwendet den Gerätenamen. Dieser ist intern der Geräte-ID zugeordnet.	any	
Alarmtyp	Identifiziert den Typ dieses Alarms, z. B. "com_cumulocity_events_TamperEvent".	string	
Nachricht	Der Text, der beim Auslösen des Alarms angezeigt wird. Die Nachricht wird entweder durch Parameter oder Eingabe festgelegt. Wenn weder das eine noch das andere angegeben ist, wird der Modellname verwendet. Sie können nicht beide angeben.	string	Optional

Name	Beschreibung	Typ	Hinweise
Schweregrad	Der Schweregrad des Alarms. Der Schweregrad wird entweder durch diesen Parameter oder dem Eingangsport festgelegt. Sie brauchen nur einen der beiden zu bestimmen.	Option - eine von: ■ Kritisch ■ Wichtig ■ Weniger wichtig ■ Warnung	Optional
Parameterfragment	Wenn dieser Parameter gesetzt ist, dann werden die eingehenden Eigenschaften am "Eigenschaften"-Eingangsport zu diesem Fragment im generierten `Alarm` hinzugefügt. Wenn er nicht gesetzt ist, werden die Eigenschaften in die oberste Ebene des `Alarm`-Parameters kopiert.	string	Optional
Asynchrone Ausgabe erzeugen	Asynchrones Erzeugen oder Löschen von Alarmen erlauben. Asynchrone Ausgabeereignisse sind Ereignisse, die keine Quell-Zeitstempel haben und nur dann von einem anderen Modell zeitasynchron verwendet werden können, wenn sie von der Plattform wieder empfangen werden.	boolean	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Alarm erzeugen	Erzeugt einen Alarm, wenn ein Signal empfangen wird.	pulse
Alarm löschen	Setzt den Alarmstatus auf GELÖSCHT, wenn wahr.	boolean
Schweregrad	Der Schweregrad des Alarms - sollte einer der Werte für Schweregrad sein (WARNUNG, WENIGER WICHTIG, WICHTIG, KRITISCH) oder GELÖSCHT.	string
Nachricht	Die Nachricht für den Alarm.	string
Zeit	Legt den Zeitstempel des Alarms fest. Wenn keine Verbindung besteht, wird die aktuelle Modellzeit verwendet.	float
Eigenschaften	Die Eigenschaften, die für den Alarm festgelegt werden sollen.	any

E-Mail senden

`apama.analyticskit.blocks.cumulocity.Send_Email`

Sendet eine E-Mail an die angegebenen E-Mail-Adressen.

Der Betreff und der Text müssen entweder über einen Parameter oder über einen Eingangsport angegeben werden, nicht über beide.

Hinweis: Bei der Ausführung im Simulations- oder Testmodus wird keine E-Mail gesendet, stattdessen wird die Ausgabe des Blocks protokolliert.

Parameter

Name	Beschreibung	Typ	Hinweise
Betreff	Der Betreff der E-Mail.	string	Optional
Text	Der Text der E-Mail.	string	Optional
Antwort an	Die Antwortadresse für die E-Mail.	string	Optional
An	Die Empfänger der E-Mail. Eine oder mehrere durch Komma getrennte E-Mail-Adressen.	string	Optional
CC	(Kopie) Die Empfänger, die eine Kopie der E-Mail erhalten sollen. Eine oder mehrere durch Komma getrennte E-Mail-Adressen.	string	Optional
BCC	(Blindkopie) Die Empfänger, die eine Blindkopie der E-Mail erhalten sollen. Eine oder mehrere durch Komma getrennte E-Mail-Adressen. Empfänger, die in An und CC aufgeführt sind, können die BCC-Liste nicht sehen.	string	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Senden	Sendet eine E-Mail, wenn ein Signal erhalten wird.	pulse
Betreff	Betreff der E-Mail.	string
Text	Text der E-Mail.	string

Ereignisausgabe

`apama.analyticskit.blocks.cumulocity.CreateEvent`

Erstellt ein neues `Event`-Objekt für das verbundene Gerät oder das Trigger-Gerät.

Dieser Block sendet ein neues Ereignis an das aktuelle oder festgelegte Gerät. Der Text des Ereignisses wird vom Eingangsport "Texteingabe" oder dem Parameter "Nachricht" bestimmt. Sie dürfen nicht beide festlegen. Ohne Angabe wird der Modellname als Text verwendet.

Dieser Block erzeugt eine synchrone Ausgabe. Die Parameter, die den Ausgabe-Stream definieren, sind "Gerät oder Trigger-Gerät" und "Ereignistyp". Die Verwendung einer anderen Instanz dieses Blocks mit demselben Ausgabe-Stream ist nicht erlaubt.

Parameter

Name	Beschreibung	Typ	Hinweise
Gerät oder Trigger-Gerät	Das Gerät (oder bei Modellen, die mit Gruppen arbeiten, das Trigger-Gerät), an das das Ereignis gesendet werden soll. Der Modelleditor verwendet den aktuellen Gerätenamen. Dieser ist intern der Geräte-ID zugeordnet.	any	
Ereignistyp	Kennzeichnet den Typ dieses Ereignisses.	string	
Nachricht	Der Text, der beim Anlegen des Ereignisses angezeigt wird. Dazu darf der Eingangsport "Texteingabe" nicht verbunden sein. Ohne Angabe wird der Modellname als Text verwendet.	string	Optional
Parameterfragment	Wenn dieser Parameter gesetzt ist, dann werden die eingehenden Eigenschaften am "Eigenschaften"-Eingangsport zu diesem Fragment im generierten `Event` hinzugefügt. Wenn er nicht gesetzt ist, werden die Eigenschaften in die oberste Ebene des `Event`-Parameters kopiert.	string	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Ereignis anlegen	Erzeugt ein Ereignis, wenn ein Signal empfangen wird.	pulse
Texteingabe	Legt den Text des Ereignisses fest. Falls dies verwendet wird, darf der Parameter "Nachricht" nicht gesetzt sein.	string

Name	Beschreibung	Typ
Zeit	Legt den Zeitstempel des Ereignisses fest. Wenn keine Verbindung besteht, wird die aktuelle Modellzeit verwendet.	float
Eigenschaften	Die Eigenschaften, die für das Ereignis festgelegt werden sollen.	any

Managed-Object-Ausgabe

`apama.analyticskit.blocks.cumulocity.ManagedObjectOutput`

Aktualisiert ein `ManagedObject`-Objekt für ein festgelegtes Gerät oder für das aktuelle Gerät.

Wenn der Parameter "Eigenschaftsname" angegeben ist, wird der Eingangsport "Wert" (oder seine Eigenschaften, wenn er einen leeren Wert hat) verwendet, um diese Eigenschaft auf dem Managed Objekt einzustellen.

Wenn der Parameter "Eigenschaftsname" nicht gesetzt ist, werden alle Eigenschaften aus dem Eingangsport "Wert" zur Aktualisierung des verwalteten Objekts verwendet.

Hinweis: Die folgenden Referenzeigenschaften eines Managed Object können mit diesem Block nicht aktualisiert werden: `childDeviceIds`, `childAssetIds`, `deviceParentIds` und `assetParentIds`.

Dieser Block ist nicht an der zeitsynchronen Modell-zu-Modell-Kommunikation beteiligt. Mehrere Blöcke können in einem einzelnen Modell oder mehreren Modellen zum Aktualisieren ein und derselben Eigenschaft ein und desselben Geräts verwendet werden. Aufgrund dieses Blocks können Kreisläufe zwischen den Modellen erzeugt werden; deshalb ist darauf zu achten, dass keine unendliche Kreislauf-Ausführung von Modellen herbeigeführt wird.

Dieser Block erzeugt eine asynchrone Ausgabe. Der Parameter, der den Ausgabe-Stream des Blocks definiert ist "Gerät" oder "Gerätegruppe".

Parameter

Name	Beschreibung	Typ	Hinweise
Gerät oder Trigger-Gerät	Das Gerät (oder bei Modellen, die mit Gruppen arbeiten, das Trigger-Gerät), mit dem das Managed Object verbunden ist. Der Modelleditor verwendet den aktuellen Gerätenamen. Dieser ist intern der Geräte-ID zugeordnet.	any	
Eigenschafts-name	Der Name der zu aktualisierenden Eigenschaft.	string	Optional

Name	Beschreibung	Typ	Hinweise
	Wenn nicht festgelegt, werden die Eigenschaften vom Eingangsport "Wert" den Eigenschaften der obersten Ebene des Managed Object zugeordnet.		

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der Wert der festzulegenden Eigenschaft.	any
Eigenschaft aktualisieren	Signalisiert, dass die Eigenschaft aktualisiert werden soll. Wenn nicht verbunden, aktualisiert jeder neue Wert aus dem Eingangsport "Wert" die Eigenschaft.	pulse

Messwertausgabe

`apama.analyticskit.blocks.cumulocity.CreateMeasurement`

Erstellt ein neues `Measurement`-Objekt für das verbundene Gerät oder das Trigger-Gerät.

Dieser Block sendet einen neuen Messwert für den "Wert"-Eingangsport an ein Gerät oder eine Gerätegruppe. Wenn der "Senden"-Eingangsport verbunden ist, wird nur dann eine Ausgabe gesendet, wenn ein Sendesignal empfangen wird. Der Messwert wird an das aktuelle oder das angegebene Gerät gesendet.

Dieser Block erzeugt eine synchrone Ausgabe. Die Parameter, die den Ausgabe-Stream des Blocks definieren, sind "Gerät oder Gerätegruppe", "Fragmentname und "Name der Serie". Die Verwendung einer anderen Instanz des Blocks mit demselben Ausgabe-Stream ist nicht erlaubt.

Unendliche Werte werden ignoriert.

Parameter

Name	Beschreibung	Typ	Hinweise
Gerät oder Trigger-Gerät	Das Gerät (oder bei Modellen, die mit Gruppen arbeiten, das Trigger-Gerät), an das der Messwert gesendet werden soll. Der Modelleditor verwendet den aktuellen Gerätenamen. Dieser ist intern der Geräte-ID zugeordnet.	any	
Fragmentname	Der Name des Fragments in der Messung.	string	
Name der Serie	Der Name der Serie in der Messung.	string	

Name	Beschreibung	Typ	Hinweise
Unit	Der Name der Maßeinheit (zum Beispiel "mm" oder "lux").	string	Optional
Parameterfragment	Wenn dieser Parameter gesetzt ist, dann werden die eingehenden Eigenschaften am "Eigenschaften"-Eingangsport zu diesem Fragment im generierten `Measurement` hinzugefügt. Wenn er nicht gesetzt ist, werden die Eigenschaften in die oberste Ebene des `Measurement`-Parameters kopiert.	string	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der zu sendende Messwert.	float
Senden	Signalisiert, dass ein neuer Messwert erstellt werden soll. Wenn nicht verbunden, erzeugt jede neue Werteingabe eine Messung.	pulse
Zeit	Legt den Zeitstempel der Messung fest. Wenn keine Verbindung besteht, wird die aktuelle Modellzeit verwendet.	float
Eigenschaften	Die Eigenschaften, die für die Messung festgelegt werden sollen.	any

Operationsausgabe

`apama.analyticskit.blocks.cumulocity.CreateOperationStaticValue`

Erstellt ein neues `Operation`-Objekt für ein angegebenes oder für das aktuelle Gerät.

Wenn keiner der Parameter "Name der Operation", "Parametername" oder "Parameterwert" gesetzt ist, dann verwendet die Operation den Eingangsport "Eigenschaften", um das `Operation`-Objekt zu füllen.

Der Block ist nicht an der zeitsynchronen Modell-zu-Modell-Kommunikation beteiligt. Mehrere Blöcke können in einem einzelnen Modell oder mehreren Modellen zum Anlegen neuer Operationen für ein und dasselbe Gerät verwendet werden. Aufgrund dieses Blocks können Kreisläufe zwischen den Modellen erzeugt werden; deshalb ist darauf zu achten, dass keine unendliche Kreislauf-Ausführung von Modellen herbeigeführt wird.

Dieser Block erzeugt eine asynchrone Ausgabe. Der Parameter, der den Ausgabe-Stream des Blocks definiert ist "Gerät" oder "Trigger-Gerät".

Parameter

Name	Beschreibung	Typ	Hinweise
Gerät oder Trigger-Gerät	Das Gerät (oder bei Modellen, die mit Gruppen arbeiten, das Trigger-Gerät), an das die Operation gesendet werden soll. Der Modelleditor verwendet den Gerätenamen. Dieser ist intern der Geräte-ID zugeordnet.	any	
Name der Operation	Der Name der Eigenschaft, die beim Objekt `Operation` erstellt werden soll.	string	Optional
Parametername	Der Name des Parameters für die Operation.	string	Optional
Parameterwert	Der Wert des Parameters für die Operation.	string	Optional
Beschreibung	Die Beschreibung der zu erstellenden Operation.	string	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Operation erstellen	Erstellt eine Operation, wenn ein Signal erhalten wird.	pulse
Eigenschaften	Die für die Operation festgelegten Eigenschaften.	any

SMS senden

`apama.analyticskit.blocks.cumulocity.Send_SMS`

Sendet eine SMS (Short Message Service) an die angegebene Telefonnummer.

Wenn der Eingangsport "SMS senden" verbunden ist, sendet dieser Block bei Empfang eines Signals eine Ausgabe an die angegebene Rufnummer.

Hinweis: Bei der Ausführung im Simulations- oder Testmodus wird keine SMS gesendet, stattdessen wird die Ausgabe des Blocks protokolliert.

Parameter

Name	Beschreibung	Typ	Hinweise
Telefonnummer	Die Telefonnummer, an die die SMS gesendet werden soll.	string	

Name	Beschreibung	Typ	Hinweise
Text	Der Inhalt der SMS. Die maximale Länge beträgt 160 Zeichen. Geben Sie diesen Parameter nicht an, wenn Sie den Eingangsport "SMS-Text " verwenden.	string	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
SMS senden	Löst das Senden einer SMS aus.	pulse
SMS-Text	Inhalt der SMS. Wenn nicht verbunden, geben Sie den Inhalt im Parameter "Text" an.	string

Logik

NICHT

`apama.analyticskit.blocks.core.Not`

Führt ein logisches 'NICHT' für die Eingabe aus.

Wenn die verbundene Eingabe `true` ist, dann ist die Ausgabe `false`. Wenn die verbundene Eingabe `false` ist, dann ist die Ausgabe `true`.

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Eingabewert für den Block.	boolean

Details zum Ausgangsport

Name	Beschreibung	Typ
NICHT	Das logische 'NICHT' der Eingabe.	boolean

ODER

`apama.analyticskit.blocks.core.Or`

Führt ein logisches 'ODER' für die Eingaben aus.

Wenn eine der verbundenen Eingaben `true` ist, dann ist die Ausgabe `true`, andernfalls `false`. Wenn alle Eingaben den Typ `pulse` haben, dann hat auch die Ausgabe den Typ `pulse`. Eingaben müssen denselben Typ haben. Der Block funktioniert auch dann, wenn einige Eingaben nicht verbunden sind.

Details zum Eingangsport

Name	Beschreibung	Typ
Wert 1	Erster Eingabewert für den Block.	boolean
Wert 2	Zweiter Eingabewert für den Block.	boolean
Wert 3	Dritter Eingabewert für den Block.	boolean
Wert 4	Vierter Eingabewert für den Block.	boolean
Wert 5	Fünfter Eingabewert für den Block.	boolean

Details zum Ausgangsport

Name	Beschreibung	Typ
ODER	Das logische 'ODER' der Eingaben.	sameAsAll(value1, value2, value3, value4, value5)

UND

`apama.analyticskit.blocks.core.And`

Führt ein logisches 'UND' für die Eingaben aus.

Wenn alle verbundenen Eingaben `true` sind, dann ist die Ausgabe `true`, ansonsten ist sie `false`.Wenn eine der Eingaben vom Typ `pulse` ist, dann ist die Ausgabe auch vom Typ `pulse`. Es sind beliebige Kombinationen von Eingaben vom Typ `pulse` und `boolean` zulässig. Der Block funktioniert auch dann, wenn einige Eingaben nicht verbunden sind.

Details zum Eingangsport

Name	Beschreibung	Typ
Wert 1	Erster Eingabewert für den Block.	boolean
Wert 2	Zweiter Eingabewert für den Block.	boolean
Wert 3	Dritter Eingabewert für den Block.	boolean

Name	Beschreibung	Typ
Wert 4	Vierter Eingabewert für den Block.	boolean
Wert 5	Fünfter Eingabewert für den Block.	boolean

Details zum Ausgangsport

Name	Beschreibung	Typ
UND	Das logische 'UND' für die Eingaben.	pulseOrBoolean(value1, value2, value3, value4, value5)

Berechnung

Ausdruck

`apama.analyticskit.blocks.core.Expression`

Bewertet einen Ausdruck, um arithmetische oder logische Berechnungen oder Zeichenfolgen-Operationen durchzuführen.

Bei Änderung der Eingabewerte (wenn alle verwendeten Eingaben erhalten wurden), wird der im Parameter angegebene Ausdruck neu berechnet.

Die Ausdruckssprache ist ähnlich wie EPL (siehe EPL-Referenz in der Dokumentation), sie ist jedoch auf die Typen `float`, `integer`, `string` und `boolean` beschränkt. Alle numerischen Literale werden als Wert vom Typ `float` behandelt, auch wenn sie keinen Bruchteil haben. Integer-Werte können nur als Ergebnis von Funktionen wie etwa `floor()` erhalten werden. Ähnlich wie bei EPL, können `integer` und `float` nicht implizit in einem Ausdruck konvertiert werden. Wenn das Ergebnis eines Ausdrucks ein `integer`-Wert ist, wird er automatisch in `float` umgewandelt (dies kann zu einem Genauigkeitsverlust führen). Boolesche Werte können mittels der booleschen Literale `true` und `false` angegeben werden. Bei booleschen Literalen wird nicht zwischen Groß- und Kleinbuchstaben unterschieden, somit ist zum Beispiel `TRUE` ebenso zulässig wie `True`. Zeichenfolgenwerte können angegeben werden, indem die Zeichenfolgen-Literale in doppelte Anführungszeichen gesetzt werden (zum Beispiel "mein Wert"). Sonderzeichen werden mit einem Rückwärtsschrägstrich (\) codiert. Die folgenden Sonderzeichen (und ihre Codierung) werden in Zeichenfolgen-Literalen unterstützt:

- Doppelte Anführungszeichen - `\"`
- Rückwärtsschrägstrich - `\\`
- Zeilenumbruch - `\n`
- Tabulator - `\t`

Die Werte der Eingaben sind als `input1`, `input2`, `input3`, `input4` und `input5` verfügbar. Die Eingabewerte können vom Typ `float`, `string`, `boolean` und `any` sein. Logische, relationale, numerische und Gleichheits-Operatoren können mit Werten der unterstützten Typen verwendet werden. Bei logischen Operatoren wird nicht zwischen Groß- und Kleinbuchstaben unterschieden, somit ist zum Beispiel `AND` ebenso zulässig wie `And`. Integrierte Methoden für die Typen `float`, `integer`, `string` und `boolean` können aufgerufen werden, unter anderem `x.abs()` (absoluter Wert von `x`), `x.pow(y)` (`x` hoch `y`), `x.sin()` (Sinus von `x` in Radiant), `x.round()` (Rundung von `x` auf die nächste Integerzahl) und `s.ltrim()` (ohne Leerzeichen am Anfang der Zeichenfolge `s`). Integrierte statische Methoden der unterstützten Typen können durch Angabe des Typnamens, gefolgt von einem Punkt (`.`) und vom Methodennamen aufgerufen werden, zum Beispiel `float.max(input1, input2)` (findet den größeren von zwei Eingabewerten). Integrierte Konstanten bei den unterstützten Typen können durch Angabe des Typnamens, gefolgt von einem Punkt (`.`) und dem Konstantennamen aufgerufen werden, zum Beispiel `float.E` (Eulersche Konstante). Werte vom Typ `any` werden zur Laufzeit entpackt, um den Ausdruck zu bewerten. Nach dem Entpacken muss der Wert vom Typ `float`, `string` oder `boolean` sein. Der Typ-Checker versucht, die Ausdrücke während der Validierungsphase zu validieren, aber dies ist beim Typ `any` nicht immer möglich. Wenn also ein Ausdruck den Typ `any` enthält, kann er, selbst wenn er die Validierungsphase übersteht, zur Laufzeit fehlschlagen, weil ein falscher Variablentyp übergeben oder eine nicht unterstützte Operation ausgeführt wurde. Eine vollständige Liste der integrierten Methoden und Konstanten finden Sie in der API-Referenz für EPL (ApamaDoc), die über die Dokumentation aufgerufen werden kann.

Beispiele:

- Umrechnung von Fahrenheit in Celsius: `(input1 - 32) * 5/9`
- Umrechnung von Tagen in Sekunden: `input1 * 86400`
- Durchschnittswert aus 4 Eingaben: `(input1 + input2 + input3 + input4) / 4`
- Vergleich mit Schwelle: `input1 > 3.1412` (siehe aber auch Schwellenblock)
- Pythagoras zur Berechnung der Hypotenuse eines rechtwinkligen Dreiecks: `(input1.pow(2) + input2.pow(2)).sqrt()`
- Vergleich auf 3 Dezimalstellen: `(input1 * 1000 - (input1 * 1000).fractionalPart()) = (input2 * 1000 - (input2 * 1000).fractionalPart())`
- Überprüfung eines Bereichs: `input1 >= 1 and input1 <= 10`
- Vergleich einer Zeichenfolge: `input1 = "my value"`
- Größerer Wert: `float.max(input1, float.PI)`
- Rest von Integerdivision: `input1.round() % input2.round()`

Parameter

Name	Beschreibung	Typ	Hinweise
Ausdruck	Ein Ausdruck - die Darstellung eines EPL-Ausdrucks als Zeichenkette.	string	

Name	Beschreibung	Typ	Hinweise
	Ähnlich EPL-Ausdrücken, aber mit den oben beschriebenen Unterschieden.		

Details zum Eingangsport

Name	Beschreibung	Typ
input1	Erster Eingabewert; im Ausdruck als `input1` zu verwenden.	any
input2	Zweiter Eingabewert; im Ausdruck als `input2` zu verwenden.	any
input3	Dritter Eingabewert; im Ausdruck als `input3` zu verwenden.	any
input4	Vierter Eingabewert; im Ausdruck als `input4` zu verwenden.	any
input5	Fünfter Eingabewert; im Ausdruck als `input5` zu verwenden.	any

Details zum Ausgangsport

Name	Beschreibung	Typ
Ergebnis	Ergebnis des Ausdrucks.	any

Bereich

`apama.analyticsbuilder.blocks.Range`

Vergleicht den Eingabewert mit den definierten unteren und oberen Bereichswerten, um zu erkennen, ob die Eingabe innerhalb oder außerhalb des Bereichs liegt oder ob sie den Bereich überschreitet.

Standardmäßig umfasst der Bereich den Wert für den unteren Bereich, schließt aber den Wert für den oberen Bereich aus. Wenn der untere Bereich z. B. 100 und der obere Bereich 200 ist, dann liegen alle Werte von 100 bis 199 innerhalb des Bereichs. 200 wird als außerhalb des Bereichs liegend betrachtet.

Ein Pulse wird gesendet, wenn der definierte Bereich überschritten wird. Das heißt, wenn entweder der untere oder der obere Bereich überschritten wird, oder wenn der Wert von unterhalb des Bereichs nach oberhalb des Bereichs (oder umgekehrt) geht, ohne jemals innerhalb des Bereichs zu liegen.

Parameter

Name	Beschreibung	Typ	Hinweise
Unterer Bereich	Der untere Bereichswert.	float	Optional
Unteren Bereich einschließen	Wenn ausgewählt, wird ein Eingabewert berücksichtigt, der dem unteren Bereich entspricht, um festzustellen, ob er innerhalb oder außerhalb des Bereichs liegt. Wenn nicht ausgewählt, wird ein Eingabewert, der dem unteren Bereich entspricht, nicht berücksichtigt.	boolean	Standard: true
Oberer Bereich	Der obere Bereichswert.	float	Optional
Oberen Bereich ausschließen	Wenn gewählt, wird ein Eingangswert, der gleich dem oberen Bereich ist, nicht berücksichtigt, um zu erkennen, ob er innerhalb oder außerhalb des Bereichs liegt. Wenn nicht ausgewählt, wird ein solcher Eingangswert herangezogen, um zu identifizieren, ob er innerhalb oder außerhalb des Bereichs liegt.	boolean	Standard: true

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der Eingabewert, der mit dem definierten Bereich abgeglichen werden soll.	float
Zurücksetzen	Setzt den Status des Blocks zurück.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Außerhalb des Bereichs	Wird auf `true` gesetzt, wenn der Eingabewert nicht innerhalb des definierten Bereichs liegt.	boolean
Im Bereich	Wird auf `true` gesetzt, wenn sich der Eingabewert innerhalb des definierten Bereichs befindet.	boolean
Überschritten	Sendet ein Pulse, wenn der Bereich überschritten wird.	pulse

Bereichssuche

`apama.analyticskit.blocks.core.RangeLookup`

Findet den Bereich, in dem der Eingabewert liegt.

Bereiche werden mit einer Liste eindeutiger Obergrenzen in aufsteigender Reihenfolge definiert. Die untere Grenze des ersten Bereichs wird vom Wert des Parameters "Mindestwert" bestimmt. Die untere Grenze jedes darauf folgenden Bereichs wird von der oberen Grenze des vorherigen Bereichs definiert. Zu welchem Bereich ein Eingabewert, der einem Grenzwert entspricht, gehört, hängt vom Parameter "Oberen Wert ausschließen" ab. Der Block gibt den zugeordneten Wert für den Bereich, in dem der Eingabewert liegt, aus oder markiert ihn als fehlgeschlagen, wenn der Eingabewert in keinem Bereich liegt.

Parameter

Name	Beschreibung	Typ	Hinweise
Mindestwert	Die untere Grenze des ersten Bereichs (erste Zeile). Wenn nichts angegeben ist, wird negative Unendlichkeit als Minimalwert genommen.	float	Optional
Bereiche	Ein Paar aus Grenzwert und zugeordnetem Wert für die Obergrenze eines Bereichs, für den der Eingabewert und ein zugeordneter Wert ermittelt werden soll, der dann als Ausgabewert verwendet wird, wenn der Eingabewert im zulässigen Bereich liegt.	Liste des oberen Grenzwertes und zugeordneten Wertes	
Oberen Wert ausschließen	Wenn gewählt, wird ein Eingabewert, der der oberen Grenze einer Zeile entspricht, als Teil des Bereichs der nächsten Zeile verstanden. Andernfalls wird ein solcher Eingabewert als Teil des Bereichs der momentanen Zeile verstanden.	boolean	Standard: true
Typ	Der für den Ausgabewert zu verwendende Typ.	Option - eine von: ■ Float ■ Boolean ■ String	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der Eingabewert, für den der Bereich gefunden werden soll.	float

Details zum Ausgangsport

Name	Beschreibung	Typ
Zugeordnet	Der zugeordnete Wert für den Bereich, der abgeglichen wird.	any
Fehlgeschlagen	Ist `true`, wenn die Eingabe nicht in einem Bereich liegt, andernfalls `false`.	boolean

Delta

`apama.analyticskit.blocks.core.Delta`

Berechnet die Differenz zwischen aufeinander folgenden Eingabewerten.

Der Block erzeugt eine Ausgabe, nachdem er mindestens zwei Eingabewerte erhalten hat.

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Wert zur Berechnung des Deltas.	float
Zurücksetzen	Setzt den Status des Blocks zurück.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Delta-Wert	Der Delta-Wert.	float

Differenz

`apama.analyticskit.blocks.core.Difference`

Berechnet die absolute und signierte Differenz zwischen den verbundenen Eingaben.

Erzeugt nur eine Ausgabe, wenn beide Eingaben einen Wert erhalten.

Details zum Eingangsport

Name	Beschreibung	Typ
Wert 1	Erster Eingabewert für den Block.	float

Name	Beschreibung	Typ
Wert 2	Zweiter Eingabewert für den Block.	float

Details zum Ausgangsport

Name	Beschreibung	Typ
Absolute Differenz	Die absolute Differenz der Eingaben.	float
Signierte Differenz	Die signierte Differenz der Eingaben. Positiv, wenn die Eingabe für Wert 1 größer ist als die für Wert 2.	float

KPI

`apama.analyticskit.blocks.cumulocity.KPI`

Vergleicht einen Wert entweder mit einem KPI (Key Performance Indicator) oder dem Datenpunkt eines Geräts.

Dieser Block verwendet Daten aus dem Eingangsport "KPI" oder aus dem Gerät, das Datenpunkte enthält. Er extrahiert die Einheiten, die Bezeichnung sowie den roten und gelben Bereich. Die Ausgabe zeigt an, ob der Wert innerhalb des vom KPI oder Datenpunkt angegebenen roten oder gelben Bereichs liegt.

Die KPI-Eingabe kann Eigenschaften bereitstellen, üblicherweise von einem KPI-verwalteten Objekt, zu denen der rote und gelbe Bereich, die Einheit und die Beschriftung gehören. Wenn das Gerät einen Datenpunkt für das angegebene "Fragment und Serie" enthält, dann überschreiben die Werte aus dem Datenpunkt die aus dem KPI.

Parameter

Name	Beschreibung	Typ	Hinweise
Quellgerät oder Gerätegruppe	Das Gerät oder die Gerätegruppe, in denen sich ein Datenpunkt befindet. Wenn angegeben, wird dieses Gerät (üblicherweise das gleiche Gerät wie die Messquelle) daraufhin überprüft, ob es einen Datenpunkt für das angegebene Fragment und Serie enthält. Wenn es einen Datenpunkt enthält, werden die roten und gelben Bereichswerte aus dem Quellobjekt anstelle der KPI-Werte verwendet. Der Modelleditor verwendet den aktuellen Geräte- oder Assetnamen. Dieser ist intern der Inventar-ID zugeordnet.	string	Optional

Name	Beschreibung	Typ	Hinweise
Datenpunkt Fragment und Serie	Dieser Parameter muss angegeben werden, wenn der Parameter "Quellgerät oder Gerätegruppe" angegeben ist. Er gibt einen Datenpunkt aus dem Quellobjekt an. Dieser ist üblicherweise identisch mit "Fragment und Serie" der Messquelle. Datenpunkt Fragment und Serie muss als `fragment.series` angegeben werden.	string	Optional
Oberes Ende des gelben Bereichs ausschließlich	Wenn gesetzt, wird das obere Ende des gelben Bereichs als exklusiv behandelt.	boolean	Standard: false
Oberes Ende des roten Bereichs exklusiv	Wenn festgelegt, wird das obere Ende des roten Bereichs als exklusiv behandelt.	boolean	Standard: false

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Numerischer Wert zum Vergleich mit den definierten Bereichen.	float
KPI	Objekt, das die Eigenschaft `c8y_Kpi` enthält.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Rot	Wird auf `true` gesetzt, wenn der Wert im roten Bereich liegt (falls ein roter Bereich definiert ist).	boolean
Gelb	Wird auf `true` gesetzt, wenn der Wert im gelben Bereich liegt (falls ein gelber Bereich definiert ist).	boolean
Einheit	Der Einheitsname des Datenpunkts.	string
Beschriftung	Der Beschriftungsname des Datenpunkts.	string

Limit

`apama.analyticsbuilder.blocks.Limit`

Gibt einen Wert aus, der innerhalb des definierten unteren und oberen Limits bleibt.

Der Eingabewert ist begrenzt, sodass die Ausgabe die durch die Parameter "Unteres Limit" und "Oberes Limit" definierten Grenzwerte nicht überschreitet. Wenn die Eingabe einen der Grenzwerte nicht einhält, wird die Ausgabe auf den Wert des Parameters gesetzt, andernfalls wird der Wert unverändert weitergegeben.

Es muss nur ein Limit angegeben werden. In diesem Fall wird die Eingabe nur in Richtung des angegebenen Parameters begrenzt.

Parameter

Name	Beschreibung	Typ	Hinweise
Oberes Limit	Jede Eingabe überhalb dieses Wertes führt dazu, dass die Ausgabe auf diesen Wert begrenzt wird.	float	Optional
Unteres Limit	Jede Eingabe unterhalb dieses Wertes führt dazu, dass die Ausgabe auf diesen Wert begrenzt wird.	float	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der Eingabewert.	float

Details zum Ausgangsport

Name	Beschreibung	Typ
Ausgabe	Der Eingabewert, wenn er innerhalb der von den Parametern "Unteres Limit" und "Oberes Limit" definierten Grenzewerte liegt. Wenn der Eingabewert einen der Grenzwertparameter überschreitet, wird er auf den Wert dieses Parameters gesetzt.	float

Machine Learning

`apama.analyticskit.blocks.core.Zementis`

Ruft das angegebene Machine Learning-Modell auf, das die Eingabedaten bewertet.

Um diesen Block zu verwenden, muss die Machine Learning-Anwendung mit den entsprechenden Machine Learning-Modellen im Mandanten zur Verfügung stehen.

Wenn das Machine Learning-Modell noch nicht vorhanden ist, fügen Sie es mithilfe der Machine Learning-Anwendung hinzu. Haben Sie das Machine Learning-Modell hinzugefügt während Ihr Analytics Builder-Modell noch im Bearbeitungsmodus war, dann verlassen Sie den Modelleditor

und bearbeiten Sie das Analytics Builder-Modell ein weiteres Mal. Dies aktualisiert die Liste aller verfügbaren Machine Learning-Modelle und Sie können das neu hinzugefügte Modell auswählen.

Blockeingänge entsprechen den Eingängen des Machine Learning-Modells (die als "Aktiv" markiert sind) in der vom Machine Learning-Modell angegebenen Reihenfolge. Alle vom Modell verwendeten Eingänge müssen verbunden sein. Ausgänge entsprechen den vom Machine Learning-Modell angegebenen Ausgänge. Wenn eine PMML-Ausgabe als "JSON" angegeben wird, gibt der Block eine String-Version des JSON aus, aber die Eigenschaften des Objekts stehen auch als zusätzliche Werte zur Verfügung, die mit dem Block "Eigenschaft extrahieren" extrahiert werden können, was die empfohlene Methode zum Entpacken mehrerer Werte aus einem solchen Ausgang ist. Zeitstempeleingaben werden derzeit nicht unterstützt.

Parameter

Name	Beschreibung	Typ	Hinweise
Machine Learning-Modell	Name des Machine Learning-Modells.	string	

Details zum Eingangsport

Name	Beschreibung	Typ
Wert 1	Eingabewert 1.	any
Wert 2	Eingabewert 2.	any
Wert 3	Eingabewert 3.	any
Wert 4	Eingabewert 4.	any
Wert 5	Eingabewert 5.	any
Wert 6	Eingabewert 6.	any
Wert 7	Eingabewert 7.	any
Wert 8	Eingabewert 8.	any
Wert 9	Eingabewert 9.	any
Wert 10	Eingabewert 10.	any

Details zum Ausgangsport

Name	Beschreibung	Typ
Ausgabe 1	Ausgabewert 1.	any
Ausgabe 2	Ausgabewert 2.	any

Name	Beschreibung	Typ
Ausgabe 3	Ausgabewert 3.	any
Ausgabe 4	Ausgabewert 4.	any
Ausgabe 5	Ausgabewert 5.	any
Ausgabe 6	Ausgabewert 6.	any
Ausgabe 7	Ausgabewert 7.	any
Ausgabe 8	Ausgabewert 8.	any
Ausgabe 9	Ausgabewert 9.	any
Ausgabe 10	Ausgabewert 10.	any

Nach Basis N

`apama.analyticsbuilder.blocks.ToBaseN`

Wandelt einen Float in einen Basis N-String um.

Der Ausgabe-String kann in einer beliebigen ganzzahligen Basis von 2 bis 36 vorliegen, dabei werden Buchstaben des deutschen Alphabets als Ziffern für Basen über 10 verwendet. Übliche Basen sind 2 (binär), 8 (oktal), 10 (dezimal) und 16 (hexadezimal). Die zu konvertierende Zahl kann einen Radix-Punkt enthalten. Die Ausgabe wird mit einer maximalen Genauigkeit von 16 Radix-Stellen berechnet.

Die Konvertierung zwischen zwei beliebigen Basen kann durch Verkettung dieses Blocks mit dem Block "Von Basis N" erfolgen.

Parameter

Name	Beschreibung	Typ	Hinweise
Ausgabebasis	Die Zahlenbasis des Ausgabe-Streams, im Bereich von 2 bis 36.	integer	Standard: 16
Radix-Zeichen	Das Zeichen, das als Radixpunkt verwendet wird. Erwartet wird ein Punkt oder ein Komma.	Option - eine von: ■ Punkt ■ Komma	Standard: Punkt

Details zum Eingangsport

Name	Beschreibung	Typ
Float	Numerische Eingabe.	float

Details zum Ausgangsport

Name	Beschreibung	Typ
Basis N	String-Ausgabe in Basis N.	string

Richtungserkennung

`apama.analyticskit.blocks.core.DirectionDetector`

Erkennt, ob der Eingabewert die Richtung ändert.

Gibt den Richtungswechsel und den letzten signifikanten Wendepunkt aus. Kleinere Abweichungen werden ignoriert, wenn die Änderungen geringer als der definierte Hysterese-Wert sind. Wiederholte Eingaben des gleichen Wertes werden ignoriert.

Parameter

Name	Beschreibung	Typ	Hinweise
Hysterese	Zählt eine Richtungsänderung nur, wenn sich der Eingabewert nach der Richtungsänderung um den definierten Hysterese-Wert ändert. Muss entweder null sein (das heißt alle Richtungsänderungen werden gezählt) oder eine positive Zahl.	float	Standard: 0.0

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Numerischer Wert, für den die Richtungsänderung erkannt werden soll.	float
Zurücksetzen	Setzt den Status des Blocks zurück.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Aufwärts	Ist `true`, wenn sich der Eingabewert in die Aufwärtsrichtung verändert, andernfalls `false`.	boolean
Wendepunkt	Der letzte vom Block erkannte Wendepunkt.	float

Rundung

`apama.analyticskit.blocks.core.Rounding`

Rundet den Eingabewert unter Verwendung einer wählbaren Regel auf eine bestimmte Anzahl an Dezimalstellen oder auf eine Ganzzahl.

Beim Runden eines numerischen Wertes wird ein Wert durch einen anderen ersetzt, der ungefähr gleich ist, aber kürzer und einfacher dargestellt werden kann.

Verfügbare Regeln:

- Beim Aufrunden (in Richtung positive Unendlichkeit) wird die Eingabe zur nächstgelegenen Zielzahl aufgerundet.
- Beim Abrunden (in Richtung negative Unendlichkeit) wird die Eingabe zur nächstgelegenen Zielzahl abgerundet.
- Beim Runden gegen Null (weg von der Unendlichkeit) wird die Eingabe in Richtung Null auf die nächstgelegene Zielzahl gerundet.
- Beim Runden auf die nächste Zahl (halb abrunden, halb in Richtung positive Unendlichkeit) wird auf die nächstgelegene Zielzahl gerundet. Bei gleichem Abstand zu den zwei nächstgelegenen Zielzahlen wird immer aufgerundet. Beispiel: Der Wert `23.5` wird auf `24` abgerundet, aber der Wert `-23.5` wird auf `-23.` gerundet.
- Beim symmetrischen Runden wird auf die nächstgelegene Zielzahl gerundet. Bei gleichem Abstand zu den zwei nächstgelegenen Zielzahlen wird immer auf die nächste gerade Zielzahl gerundet. Beispiel: `0.5` wird auf `0` abgerundet und `1.5` wird auf `2.` aufgerundet. Dies nennt man auch mathematisches Runden.

Der Wert wird auf die nächstgelegene 'Zielzahl' gerundet. Das ist entweder eine Ganzzahl (eine Zahl ohne Dezimalstellen) oder eine Zahl mit einer bestimmten Anzahl an Dezimalstellen. Wenn die Anzahl der Dezimalstellen negativ ist, wird auf hoch 10 gerundet. Beispiel: Wenn die Anzahl der Dezimalstellen 2 ist, wird auf das nächstgelegene Hundertstel (0,01) gerundet. Wenn die Anzahl der Dezimalstellen -3 ist, wird auf die nächstgelegenen Tausend (1000) gerundet.

Parameter

Name	Beschreibung	Typ	Hinweise
Regel	Die anzuwendende Rundungsregel.	Option - eine von: ■ Aufwärts ■ Nach unten ■ Gegen Null ■ Auf nächsten Wert ■ Symmet-risch	
Anzahl der Dezimalstellen	Die Anzahl der Dezimalstellen, auf die der Eingabewert gerundet werden soll.	integer	Standard: 0

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der zu rundende Eingabewert.	float

Details zum Ausgangsport

Name	Beschreibung	Typ
Gerundeter Wert	Der auf eine bestimmte Anzahl an Dezimalstellen oder auf eine Ganzzahl gerundete Eingabewert.	float

Schwelle

`apama.analyticskit.blocks.core.Threshold`

Vergleicht den Eingabewert mit dem definierten Schwellenwert, um zu erkennen, ob der Eingabewert den Schwellenwert durchbricht oder überschreitet.

Der Schwellenwert ist dann durchbrochen, wenn die Richtung als 'Oberhalb' bestimmt wurde und der Eingabewert größer als der festgelegte Schwellenwert ist oder wenn die Richtung als 'Unterhalb' bestimmt wurde und der Eingabewert kleiner als der festgelegte Schwellenwert ist oder wenn die Richtung als 'Oberhalb oder gleich' bestimmt wurde und der Eingabewert größer als oder gleich dem festgelegten Schwellenwert ist oder wenn die Richtung als 'Unterhalb oder

gleich' bestimmt wurde und der Eingabewert kleiner als oder gleich dem festgelegten Schwellenwert ist.

Ein Pulse wird gesendet, wenn der festgelegte Schwellenwert aus jeder Richtung überquert wird.

Parameter

Name	Beschreibung	Typ	Hinweise
Schwellenwert	Dieser Wert wird mit dem Eingabewert verglichen.	float	
Richtung	Die Richtung, in der gesucht werden soll: ob der Eingabewert über, unter oder genau an der definierten Schwelle liegt oder ob er die definierte Schwelle überquert.	Option - eine von: ■ Oberhalb ■ Oberhalb oder gleich ■ Unterhalb ■ Unterhalb oder gleich ■ Überschreitung	Standard: Oberhalb

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der Eingabewert, der mit dem definierten Schwellenwert verglichen werden soll.	float
Zurücksetzen	Setzt den Status des Blocks zurück.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Schwelle durchbrochen	Wird auf `true` gesetzt, wenn die Schwelle durchbrochen wird.	boolean
Innerhalb Schwelle	Wird auf `true` gesetzt, wenn die Schwelle nicht durchbrochen wird.	boolean
Schwelle überschritten	Sendet ein Pulse, wenn die Schwelle überschritten wird.	pulse

Überschreitungszähler

`apama.analyticskit.blocks.core.CrossingCounter`

Erkennt und zählt die Anzahl der Schwellenüberschreitungen in der angegebenen Richtung.

Als Überschreitung gilt eine Änderung des Eingabewertes von einer Seite des Schwellenwertes auf die andere Seite (der Eingabewert ist also zuerst kleiner, dann größer als der Schwellenwert oder umgekehrt).

Der Block kann über ein zeitgebundenes Fenster wirken, das im Parameter "Fensterdauer" bestimmt wird. Wenn dieser Parameter nicht angegeben ist, wird ein nicht zeitgebundenes Fenster verwendet. Beim Zurücksetzen wird der Inhalt des Fensters gelöscht.

Wenn ein Fenster konfiguriert wurde, verwendet der Block einen Satz von 20 Containern, sodass die Zeit der abgelaufenen Werte ungefähr dem nächsten Containerintervall entspricht.

Parameter

Name	Beschreibung	Typ	Hinweise
Schwellenwert	Dieser Wert wird mit dem Eingabewert verglichen.	float	Standard: 0.5
Richtung	Die Richtung, in der nach einer Überquerung der Schwelle gesucht werden soll: aufwärts, abwärts oder in beide Richtungen.	Option - eine von: ■ Aufwärts ■ Abwärts ■ Beide	
Fensterdauer (Sek.)	Wenn vorhanden, die Zeitdauer (in Sekunden), für die Werte im Fenster bleiben sollen. Dies muss eine endliche und positive Zahl sein.	float	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der Eingabewert für den eine Überschreitung erkannt werden soll.	float
Zurücksetzen	Setzt die Anzahl der Überschreitungen zurück.	pulse
Probe	Erzwingt die Neubewertung des aktuellen Wertes und sendet den Ausgabewert.	boolean

Details zum Ausgangsport

Name	Beschreibung	Typ
Anzahl Überschreitungen	Die Anzahl der Überschreitungen.	float
Überschreitung	Sendet einen Pulse, wenn eine Überschreitung erkannt wird.	pulse

Von Basis N

`apama.analyticsbuilder.blocks.FromBaseN`

Wandelt einen Basis N-String in einen Float um.

Der Eingabe-String kann in einer beliebigen ganzzahligen Basis von 2 bis 36 vorliegen, dabei werden Buchstaben des deutschen Alphabets als Ziffern für Basen über 10 verwendet. Übliche Basen sind 2 (binär), 8 (oktal), 10 (dezimal) und 16 (hexadezimal). Die zu konvertierende Zahl kann einen Radix-Punkt enthalten.

Die Konvertierung zwischen zwei beliebigen Basen kann durch Verkettung dieses Blocks mit dem Block "Nach Basis N" erfolgen.

Parameter

Name	Beschreibung	Typ	Hinweise
Eingabebasis	Die Zahlenbasis des Eingang-Streams, im Bereich von 2 bis 36.	integer	Standard: 16
Radix-Zeichen	Das Zeichen, das als Radixpunkt verwendet wird. Erwartet wird ein Punkt oder ein Komma.	Option - eine von: ■ Punkt ■ Komma	Standard: Punkt

Details zum Eingangsport

Name	Beschreibung	Typ
Basis N	String-Eingabe in Basis N.	string

Details zum Ausgangsport

Name	Beschreibung	Typ
Float	Numerische Ausgabe.	float

Aggregat

Diskrete Statistik

`apama.analyticsbuilder.blocks.DiscreteStatistics`

Erzeugt Statistiken der Summe, Anzahl, Durchschnitt (Mittelwert), Standardabweichung, Minimum und Maximum für diskrete Eingabewerte.

Dieser Block eignet sich für Eingaben der diskreten Zeit, bei denen die Anzahl der Proben (oder Eingaben) signifikant ist, die Zeit zwischen ihnen jedoch nicht. Die Blöcke "Durchschnitt" (Mittelwert) und "Standardabweichung" eignen sich besser für kontinuierliche Werte, als solche die unregelmäßig als Stichprobe genommen werden, wie beispielsweise Temperaturwerte. Verwenden Sie diesen Block zum Beispiel, wenn jede Probe eine Transaktion darstellt, so wie etwa ein Ticketverkauf.

Wenn der Eingangsport " Probe" nicht verbunden ist, wird jeder Wert abgetastet, auch während des gleichen Aktivierungszeitraums wie Zurücksetzen. Da jeder Wert benutzt wird, verwendet die Standardabweichung die allgemeine Formel: $\sigma^2 = \sum(x - \mu)^2 / N$.

Wenn der Eingangsport "Probe" verbunden ist, tastet der Block die Daten nur ab, wenn der Eingangsport "Probe" ein Signal empfängt. In diesem Fall verwendet die Standardabweichung der Probe die Formel: $\sigma^2 = \sum(x - \mu)^2 / (N-1)$.

Wenn die Signale "Zurücksetzen" und "Probe" gleichzeitig erhalten werden, wird "Zurücksetzen" zuerst verarbeitet.

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der aktuelle Wert.	float
Probe	Verwenden Sie den aktuellen Wert zum Generieren von Statistiken. Wenn er nicht verbunden ist, werden alle Werte verwendet.	pulse
Zurücksetzen	Setzt den Status des Blocks zurück.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Summe	Summe der abgetasteten Eingabewerte, $\sum x$.	float
Anzahl	Anzahl der abgetasteten Eingabewerte., N.	float
Durchschnitt	Durchschnitt (Mittelwert) der abgetasteten Eingabewerte, $\mu = \sum x / N$.	float
Standardabweichung	Standardabweichung der Probe-Eingabewerte. Wenn alle Werte abgetastet werden und der Eingangsport "Probe" nicht verbunden ist, wird die allgemeine Standardabweichung, $\sigma^2 = \sum(x - \mu)^2 / N$, verwendet. Andernfalls wird die Standardabweichung der Probe, $\sigma^2 = \sum(x - \mu)^2 / (N-1)$, verwendet.	float
Minimum	Minimum der abgetasteten Eingabewerte.	float
Maximum	Maximum der abgetasteten Eingabewerte.	float

Durchschnitt (Mittelwert)

`apama.analyticskit.blocks.core.Mean`

Berechnet den Mittelwert der Werte im Laufe der Zeit.

Dieser Block ist für kontinuierliche Werte geeignet, auch wenn diese unregelmäßig abgetastet werden. Die Zeit zwischen den Eingaben oder Proben ist von Bedeutung, die Anzahl der Proben hingegen nicht. Verwenden Sie diesen Block z. B., wenn es sich bei der Eingabe um eine physikalische Eigenschaft, wie etwa die Temperatur, handelt, die entweder regelmäßig oder unregelmäßig abgetastet wird (z. B. Erzeugung von Messwerten lediglich bei einer Temperaturänderung). Verwenden Sie den Block "Diskrete Statistik" anstelle des Blocks "Durchschnitt (Mittelwert)" für unabhängige Messungen, wie etwa Ticketverkäufe, bei denen die Anzahl der Messungen von Bedeutung ist, die Zeit zwischen den Messungen jedoch nicht.

Der Mittelwert ist definiert als die Summe der Eingabewerte multipliziert mit der Zeitdauer, die die Eingabe diesen Wert innerhalb eines optionalen Zeitfensters hatte, dividiert durch die Fensterdauer oder die Zeit seit der Block gestartet oder zurückgesetzt wurde (je nachdem, welche kleiner ist).

Der Block kann über ein zeitgebundenes Fenster wirken, das im Parameter "Fensterdauer" bestimmt wird. Wenn dieser Parameter nicht angegeben ist, wird ein nicht zeitgebundenes Fenster verwendet. Beim Zurücksetzen wird der Inhalt des Fensters gelöscht. Ausgabe wird erzeugt bei neuen Eingaben oder, wenn der Parameter für den Ausgabeschwellenwert gesetzt ist, nur wenn sich die Ausgabe um mehr als den angegebenen Schwellenwert ändert (dazu zählt auch, wenn keine weitere Eingabe erfolgt oder sich der Wert nur ändert, weil ältere Einträge verfallen). Der Probe-Eingangsport kann verwendet werden, um eine Neubewertung zu erzwingen und den aktuellen Wert zu erzeugen.

Weitere Informationen sowie ein Beispiel zur Ausgabehäufigkeit aus diesem Block und zum Verhalten der Werte in Fenstern finden Sie im Kapitel "Wertetypen" der Dokumentation zu Analytics Builder.

Wenn ein Fenster konfiguriert wird, verwendet der Block einen Satz von 20 Containern, sodass der abgelaufene Wert ungefähr dem Durchschnittswert eines Containers entspricht.

Hinweis: Der Block "Durchschnitt (Mittelwert)" erzeugt den Mittelwert für ein einzelnes Gerät. Wenn die Eingabe von einer Gerätegruppe stammt, werden diese Werte für jedes Gerät in dieser Gruppe einzeln erzeugt. Um aggregierte Werte für die Gruppe als Ganzes (nicht für einzelne Geräte) zu berechnen und zu erzeugen, verwenden Sie den Block "Gruppenstatistik".

Parameter

Name	Beschreibung	Typ	Hinweise
Fensterdauer (Sek.)	Wenn vorhanden, die Zeitdauer (in Sekunden), für die Werte im Fenster bleiben sollen. Dies muss eine endliche und positive Zahl sein.	float	Optional
Ausgabe-schwellenwert	Wenn vorhanden, die zu sendende Ausgabe, wenn sie sich um mindestens diesen Wert ändert. Dies muss eine endliche und positive Zahl sein.	float	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Eingabe, für die der Mittelwert berechnet werden soll.	float
Zurücksetzen	Löscht den Inhalt des Fensters.	pulse
Probe	Erzwingt die Neubewertung des aktuellen Mittelwertes und sendet den Ausgabewert.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Durchschnitt	Die Summe des Wertes multipliziert mit der Dauer, wie lange dieser Wert beibehalten wird, dividiert durch die Gesamtzeit.	float

Gradient

`apama.analyticskit.blocks.core.Gradient`

Berechnet den gewichteten linearen Regressionsgradienten für die Werte.

Der Gradient misst die Veränderungsrate eines Werts über die Zeit. Ein positiver Gradient weist auf eine Erhöhung der Eingabewerte hin, ein negativer Gradient weist auf eine Verringerung der Eingabewerte hin. Der Betrag des Gradienten gibt die Höhe der Änderung an.

Der Block kann über ein zeitgebundenes Fenster wirken, das im Parameter "Fensterdauer" bestimmt wird. Wenn dieser Parameter nicht angegeben ist, wird ein nicht zeitgebundenes Fenster verwendet und der Block wird pro Sekunde neu ausgewertet und verwendet Container von 1 Sekunde. Wenn ein Fenster konfiguriert wurde, verwendet der Block einen Satz von 20 Containern, sodass die Zeit der abgelaufenen Werte ungefähr dem nächsten Containerintervall entspricht. Die erste Gradientenausgabe wird erst generiert, wenn mindestens zwei Container zur Berechnung verfügbar sind.

Die Reset-Eingabe löscht den Inhalt des Fensters. Der Eingangsport "Beispiel" kann verwendet werden, um eine Neuauswertung zu erzwingen und den aktuellsten Wert zu generieren.

Parameter

Name	Beschreibung	Typ	Hinweise
Fensterdauer (Sek.)	Wenn vorhanden, die Zeitdauer (in Sekunden), für die Werte im Fenster bleiben sollen. Dies muss eine endliche und positive Zahl sein.	float	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der Eingabewert, für den der Gradient berechnet werden soll.	float
Zurücksetzen	Löscht den Inhalt des Fensters.	pulse
Probe	Erzwingt die Neubewertung des aktuellen Wertes und sendet den Ausgabewert.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Gradient	Der Gradient der Eingabewerte.	float

Gruppenstatistik

`apama.analyticskit.blocks.core.GroupStatistics`

Erzeugt regelmäßig wiederkehrende Aggregatwerte für alle Geräte in einer Gruppe, für die der Block Eingabewerte erhalten hat.

Dieser Block generiert die folgenden Aggregatwerte:

- Minimum
- Maximum
- Geräteanzahl
- Durchschnitt
- Standardabweichung
- Varianz

Der Block kann über ein zeitgebundenes Fenster wirken, das im Parameter "Fensterdauer" bestimmt wird. Wenn dieser Parameter nicht angegeben ist, verwendet der Block ein nicht zeitgebundenes Fenster. Die Ausgabe wird regelmäßig wiederkehrend erzeugt, wie durch den Parameter "Ausgabezeitraum" festgelegt.

Wenn ein Fenster konfiguriert wurde, verwendet der Block einen Satz von 20 Containern, sodass der abgelaufene Wert ungefähr dem nächsten Containerintervall entspricht.

Hinweis: Der "Gruppenstatistik"-Block berechnet und generiert Aggregatwerte für die Gruppe als Ganzes (nicht für einzelne Geräte). Um Aggregate für ein einzelnes Gerät in einer Gruppe zu erzeugen, verwenden Sie die Blöcke "Durchschnitt (Mittelwert)", "Standardabweichung" oder "Minimum / Maximum". Der "Gruppenstatistik"-Block berücksichtigt nur Geräte, von denen er Eingabewerte erhalten hat.

Parameter

Name	Beschreibung	Typ	Hinweise
Fensterdauer (Sek.)	Wenn vorhanden, die Zeitdauer (in Sekunden), für die Werte im Fenster bleiben sollen. Dies muss eine endliche und positive Zahl sein.	float	Optional
Ausgabezeitraum (secs)	Die Zeitdauer (in Sekunden) zwischen den einzelnen Ausgaben. Dies muss eine endliche und positive Zahl sein.	float	Standard: 5.0

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Eingabewert, für den die Aggregat-Werte berechnet werden sollen.	float

Details zum Ausgangsport

Name	Beschreibung	Typ
Minimum	Der höchste Eingabewert (am nächsten zur negativen Unendlichkeit) von allen Geräten einer Gruppe.	float
Maximum	Der höchste Eingabewert (am nächsten zur positiven Unendlichkeit) von allen Geräten einer Gruppe.	float
Geräteanzahl	Die Anzahl der Geräte in einer Gruppe, für die bisher Eingabewerte empfangen wurden.	float
Durchschnitt	Die Summe des Wertes multipliziert mit der Dauer, wie lange dieser Wert beibehalten wird, dividiert durch die Gesamtzeit und die Geräteanzahl.	float
Standardabweichung	Die Standardabweichung der Eingabewerte von allen Geräten einer Gruppe.	float
Varianz	Die Varianz der Eingabewerte von allen Geräten einer Gruppe.	float

Integral

`apama.analyticskit.blocks.core.Integral`

Berechnet den Integralwert des Eingabewertes im Laufe der Zeit.

Integral ist definiert als die Summe der Eingabewerte multipliziert mit der Zeitdauer, die die Eingabe diesen Wert hatte - entweder innerhalb eines optionalen Zeitfensters oder seit der Block gestartet oder zurückgesetzt wurde.

Der Block kann über ein zeitgebundenes Fenster wirken, das im Parameter "Fensterdauer" bestimmt wird. Wenn dieser Parameter nicht angegeben ist, wird ein nicht zeitgebundenes Fenster verwendet. Beim Zurücksetzen wird der Inhalt des Fensters gelöscht. Ausgabe wird erzeugt bei neuen Eingaben oder, wenn der Parameter für den Ausgabeschwellenwert gesetzt ist, nur wenn sich die Ausgabe um mehr als den angegebenen Schwellenwert ändert (dazu zählt auch, wenn keine weitere Eingabe erfolgt oder sich der Wert nur ändert, weil ältere Einträge verfallen). Der Probe-Eingang kann verwendet werden, um eine Neubewertung zu erzwingen und den aktuellen Wert zu erzeugen.

Wenn ein Fenster konfiguriert wurde, verwendet der Block einen Satz von 20 Containern, sodass der abgelaufene Wert ungefähr dem Durchschnittswert eines Containers entspricht.

Parameter

Name	Beschreibung	Typ	Hinweise
Fensterdauer (Sek.)	Wenn vorhanden, die Zeitdauer (in Sekunden), für die Werte im Fenster bleiben sollen. Dies muss eine endliche und positive Zahl sein.	float	Optional
Ausgabe-schwellenwert	Wenn vorhanden, die zu sendende Ausgabe, wenn sie sich um mindestens diesen Wert ändert. Dies muss eine endliche und positive Zahl sein.	float	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Eingabe, für die der Integralwert berechnet werden soll.	float
Zurücksetzen	Löscht den Inhalt des Fensters.	pulse
Probe	Erzwingt die Neubewertung des aktuellen Ganzzahlenwertes und sendet den Ausgabewert.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Integralwert	Die Summe des Wertes multipliziert mit der Dauer, wie lange dieser Wert beibehalten wird.	float

Minimum / Maximum

`apama.analyticskit.blocks.core.MinMax`

Berechnet Minimum und Maximum eines Wertes im Laufe der Zeit.

Das Minimum wird als der kleinste Eingabewert im Fenster definiert (der Wert, der der negativen Unendlichkeit am nächsten liegt). Das Maximum wird als der größte Eingabewert im Fenster definiert (der Wert, der der positiven Unendlichkeit am nächsten liegt).

Der Block kann über ein zeitgebundenes Fenster wirken, das im Parameter Fensterdauer bestimmt wird. Wenn dieser Parameter nicht angegeben ist, verwendet der Block ein ungebundenes Fenster. Die Reset-Eingabe löscht den Fensterinhalt. Ausgabe wird für alle neuen Eingaben erzeugt, die den aktuellen Mindest- oder Maximalwert unter- oder überschreiten, oder, wenn der Parameter Fensterdauer angegeben ist, wenn ein vorheriger Mindest-/Maximalwert verfällt.

Wenn ein Fenster konfiguriert wurde, verwendet der Block einen Satz von 20 Containern, sodass die Zeit der abgelaufenen Werte ungefähr dem nächsten Containerintervall entspricht.

Hinweis: Der Block "Minimum /Maximum" erzeugt Minimum und Maximum eines Wertes für ein einzelnes Gerät. Wenn die Eingabe von einer Gerätegruppe stammt, werden diese Werte für jedes Gerät in dieser Gruppe einzeln erzeugt. Um aggregierte Werte für die Gruppe als Ganzes (nicht für einzelne Geräte) zu berechnen und zu erzeugen, verwenden Sie den Block "Gruppenstatistik".

Parameter

Name	Beschreibung	Typ	Hinweise
Fensterdauer (Sek.)	Wenn vorhanden, die Zeitdauer (in Sekunden), für die Werte im Fenster bleiben sollen. Dies muss eine endliche und positive Zahl sein.	float	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Eingabe, für die Minimum und Maximum berechnet werden soll.	float
Zurücksetzen	Löscht den Inhalt des Fensters.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Minimum	Der kleinste Wert im Fenster (am nächsten zur negativen Unendlichkeit).	float
Maximum	Der höchste Wert im Fenster (am nächsten zur positiven Unendlichkeit).	float

Standardabweichung

`apama.analyticskit.blocks.core.StandardDeviation`

Berechnet die Standardabweichung und die Varianz der Werte im Laufe der Zeit.

Dieser Block ist für kontinuierliche Werte geeignet, auch wenn diese unregelmäßig abgetastet werden. Die Zeit zwischen den Eingaben oder Proben ist von Bedeutung, die Anzahl der Proben hingegen nicht. Verwenden Sie diesen Block z. B., wenn es sich bei der Eingabe um eine physikalische Eigenschaft, wie etwa die Temperatur, handelt, die entweder regelmäßig oder unregelmäßig abgetastet wird (z. B. Erzeugung von Messwerten lediglich bei einer

Temperaturänderung). Verwenden Sie den Block "Diskrete Statistik" anstelle des Blocks "Durchschnitt (Mittelwert)" für unabhängige Messungen, wie etwa Ticketverkäufe, bei denen die Anzahl der Messungen von Bedeutung ist, die Zeit zwischen den Messungen jedoch nicht.

Die Standardabweichung ist ein Maß zur Quantifizierung der Abweichung eines Satzes an Datenwerten. Eine niedrige Standardabweichung weist darauf hin, dass die einzelnen Datenpunkte näher am Mittelwert des Satzes liegen, während eine hohe Standardabweichung darauf hinweist, dass die Datenpunkte über einen größeren Bereich verteilt sind.

Der Block kann über ein zeitlich begrenztes Fenster arbeiten, das mit dem Parameter "Fensterdauer" angegeben wird. Wenn dieser Parameter nicht angegeben wird, verwendet der Block ein unbegrenztes Fenster. Der Eingangsport "Zurücksetzen" löscht den Fensterinhalt. Der Eingangsport "Beispiel" kann verwendet werden, um eine Neubewertung zu erzwingen und den aktuellsten Wert zu erzeugen. Die Ausgabe wird bei jedem neuen Eingang generiert oder, falls der Parameter "Fensterdauer" festgelegt ist, wird die Ausgabe periodisch bei jedem neuen Bucket generiert, der dem Fenster hinzugefügt wird

Wenn ein Fenster konfiguriert wurde, verwendet der Block einen Satz von 20 Containern, sodass die Zeit der abgelaufenen Werte ungefähr dem nächsten Containerintervall entspricht.

Hinweis: Der "Standardabweichung"-Blockerzeugt die Standardabweichung und Varianz für ein einzelnes Gerät. Wenn die Eingabe von einer Gerätegruppe stammt, werden diese Werte für jedes Gerät in dieser Gruppe separat erzeugt. Um aggregierte Werte für die Gruppe als Ganzes (nicht für einzelne Geräte) zu berechnen und zu erzeugen, verwenden Sie den "Gruppenstatistik"-Block.

Parameter

Name	Beschreibung	Typ	Hinweise
Fensterdauer (Sek.)	Wenn vorhanden, die Zeitdauer (in Sekunden), für die Werte im Fenster bleiben sollen. Dies muss eine endliche und positive Zahl sein.	float	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Eingabe, für die Standardabweichung und Varianz berechnet werden sollen.	float
Zurücksetzen	Löscht den Inhalt des Fensters.	pulse
Probe	Erzwingt die Neubewertung des aktuellen Wertes und sendet den Ausgabewert.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Standard-abweichung	Die Standardabweichung der Eingabewerte.	float
Varianz	Die Varianz der Eingabewerte.	float

Zähler

`apama.analyticskit.blocks.core.Counter`

Zählt die Gesamteingaben und die wiederholten Eingaben.

Wenn zwei aufeinanderfolgende Eingabewerte verschiedene Typen haben, werden sie nicht als Wiederholungswerte ausgewertet. Alle anderen Auswertungen, die aufzeigen, ob zwei Werte gleich sind, verwenden die gleichen Regeln wie EPL. Weitere Informationen finden Sie im folgenden Abschnitt der Dokumentation: Developing Apama Applications > EPL Reference > Types.

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Die zu zählende Eingabe.	any
Zurücksetzen	Setzt die Zähler zurück.	boolean

Details zum Ausgangsport

Name	Beschreibung	Typ
Anzahl	Die Gesamtanzahl aller Eingaben.	float
Anzahl gleich	Die aktuelle Anzahl wiederholter Eingaben mit demselben Wert. Bei einer Änderung des Werts, wird die Ausgabe auf eins zurückgesetzt.	float

Fluss-Manipulation

Gate

`apama.analyticskit.blocks.core.Gate`

Hindert die Eingabe daran, zur Ausgabe zu werden, sofern das Gate nicht geöffnet und eingeschaltet ist.

Der Block startet als nicht aktiviert, wenn der Aktivieren-Port verbunden ist; andernfalls ist der Block immer aktiviert. Der Block startet als geschlossen, wenn der Öffnen-Port verbunden ist, andernfalls startet der Block als offen.

Parameter

Name	Beschreibung	Typ	Hinweise
Nullwert	Der Wert, der als Ausgabewert verwendet werden soll, wenn das Gate entweder nicht aktiviert oder geschlossen ist. Der genaue Typ des Wertes hängt vom Nullwerttyp-Parameter ab. Wenn der Nullwerttyp-Parameter angegeben ist, dann wird der String-Wert als der angegebene Typ erfasst. Wenn der Nullwerttyp-Parameter nicht angegeben ist und der Wert als float oder boolescher Wert geparsed werden kann, dann wird er als der jeweilige Werttyp geparsed; ansonsten wird er als String-Wert verwendet.	string	Optional
Nullwert-Typ	Der Typ des vom Nullwert-Parameter bestimmten Wertes. Wenn angegeben, dann wird der Nullwert in den angegebenen Typ geparsed.	Option - eine von: ■ Float ■ Boolean ■ String	Optional
Dauer geschlossen (Sek.)	Die Zeitdauer (in Sekunden), die das Gate geschlossen sein soll, wenn ein Schließen-Pulse empfangen wird. Wenn der Parameter nicht angegeben wird, bleibt das Gate nach dem Erhalt eines Schließen-Pulses so lange geschlossen bis es ein Öffnen-Pulse erhält. Wenn der Parameter angegeben wird, dann wird das Gate nach einer bestimmten Anzahl von Sekunden automatisch geöffnet, sofern vor Ablauf dieser Zeit kein anderes Öffnen- oder Schließen-Pulse erhalten wurde. Wenn das Gate beim Öffnen auch aktiviert ist, wird der aktuelle Eingabewert als Ausgabe gesendet. Dies muss eine endliche und positive Zahl sein.	float	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der Wert, der an die Ausgabe weitergegeben wird.	any
Aktivieren	Aktiviert das Gate. Wenn das Gate immer aktiviert sein soll, lassen Sie diesen Port unverbunden.	boolean
Öffnen	Öffnet das Gate. Wenn das Gate beim Start offen sein soll, dann lassen Sie diesen Port unverbunden.	pulse
Schließen	Schließt das Gate. Wenn der Parameter für die Zeitdauer angegeben ist, wird das Gate nur für die angegebene Dauer (in Sekunden) geschlossen.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Wert im Gate	Gleicher Wert wie der Eingabewert, wenn das Gate geöffnet und aktiviert ist. Andernfalls der vom Nullwert-Parameter bestimmte Wert.	input(value)
Eingeschaltet	Ausgabe-Pulse. Wird erzeugt, wenn das Gate eingeschaltet wird. Das Gate ist eingeschaltet, wenn es sowohl offen als auch aktiviert ist.	pulse
Ausgeschaltet	Ausgabe-Pulse. Wird erzeugt, wenn das Gate ausgeschaltet wird. Das Gate ist ausgeschaltet, wenn es entweder geschlossen oder inaktiv ist.	pulse

Kombinierer

`apama.analyticskit.blocks.core.Combiner`

Berechnet die Ausgabe auf Grundlage des gewählten Modus und der verbundenen Eingänge.

Verfügbare Modi:

- Minimum: Gibt das Minimum der verbundenen Ports aus, die einen Wert erhalten haben. Alle numerischen Literale werden als `float`-Werte behandelt.
- Maximum: Gibt das Maximum der verbundenen Ports aus, die einen Wert erhalten haben. Alle numerischen Literale werden als `float`-Werte behandelt.
- Durchschnitt (Mittelwert): Gibt den Durchschnitt (Mittelwert) der verbundenen Ports aus, die einen Wert erhalten haben. Alle numerischen Literale werden als `float`-Werte behandelt.

- Zuletzt geändert: Gibt den zuletzt geänderten Wert aus. Wenn sich mehrere Werte in einer Aktivierung gleichzeitig ändern, wird der Eingangsport mit der höchsten Nummer verwendet. Beispiel: Wenn Wert 1 und Wert 2 aktualisiert werden, wird Wert 2 für die Ausgabe gewählt. Eingaben müssen den gleichen Typ haben.

Parameter

Name	Beschreibung	Typ	Hinweise
Modus	Der zu wählende Modus.	Option - eine von: ■ Minimum ■ Maximum ■ Durchschnitt (Mittelwert) ■ Zuletzt geändert	

Details zum Eingangsport

Name	Beschreibung	Typ
Wert 1	Erster Eingabewert für den Block.	any
Wert 2	Zweiter Eingabewert für den Block.	any
Wert 3	Dritter Eingabewert für den Block.	any
Wert 4	Vierter Eingabewert für den Block.	any
Wert 5	Fünfter Eingabewert für den Block.	any

Details zum Ausgangsport

Name	Beschreibung	Typ
Kombinierter Wert	Der berechnete kombinierte Wert.	sameAsAll(value1, value2, value3, value4, value5)

Latch-Werte

`apama.analyticskit.blocks.core.Latch`

Rastet den aktuellen Eingabewert ein, der empfangen wurde, während der Block eingeschaltet ist.

Erzeugt nur dann eine Ausgabe, wenn sich der Eingabewert ändert und der Block eingeschaltet ist. Der Block startet als ausgeschaltet, wenn der Einschalten-Port verbunden ist, andernfalls ist der Block immer eingeschaltet.

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der zu überwachende Eingabewert.	any
Einschalten	Schaltet den Block ein. Wenn der Block immer eingeschaltet sein soll, lassen Sie dies unverbunden.	boolean

Details zum Ausgangsport

Name	Beschreibung	Typ
Latch-Wert	Gleicher Wert wie der Eingabewert. Wird nur erzeugt, wenn sich die Eingabe geändert hat, während der Block eingeschaltet war.	input(value)
Geändert	Ausgabe-Pulse. Wird nur erzeugt, wenn sich die Eingabe geändert hat, während der Block eingeschaltet war.	pulse
Eingeschaltet	Ausgabe-Pulse. Wird erzeugt, wenn der Block eingeschaltet ist.	pulse
Ausgeschaltet	Ausgabe-Pulse. Wird erzeugt, wenn der Block ausgeschaltet ist.	pulse

Pulse

`apama.analyticskit.blocks.core.Pulse`

Wandelt eine Nicht-Pulse-Eingabe in eine Pulse-Ausgabe um.

Das ist nützlich bei Blöcken, die sowohl Pulse- als auch Nicht-Pulse-Werte verarbeiten und bei denen der Eingangswert ohne entsprechende Umrechnung nicht als Pulse behandelt wird.

Beispielsweise wird ein numerischer Wert, der an den ODER-Block weitergegeben wird, als `true` behandelt, wenn er ungleich Null ist (wie im Abschnitt "Typumwandlungen" der Dokumentation zu Analytics Builder beschrieben). Wenn Sie jedoch einen numerischen Wert an den Pulse-Block weitergeben und dann den Ausgang des Pulse-Blocks mit dem ODER-Block verbinden, wird der numerische Wert in einen Pulse umgewandelt, sodass der ODER-Block einen Pulse sendet. Der Block kann so konfiguriert werden, dass er einen Pulse sendet, wenn sich der Wert ändert

(Standardverhalten der Pulse-Umwandlung), bei jeder Eingabe oder bei jedem Wert ungleich Null.

Parameter

Name	Beschreibung	Typ	Hinweise
Modus	Regelt, wann der Block einen Puls sendet.	Option - eine von: ■ Bei jeder Änderung ■ Bei jeder Eingabe ■ Bei Werten ungleich Null	Standard: Bei jeder Änderung

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Die als Pulse zu behandelnde Eingabe.	any

Details zum Ausgangsport

Name	Beschreibung	Typ
Pulse	Der aus dem Eingabewert konvertierte Ausgabe-Pulse-Wert.	pulse

Schalter

`apama.analyticsbuilder.blocks.Switch`

Gibt die Werte aus einem gegebenen Eingang aus oder dient als Schutzschalter.

Wenn der Parameter "Gewählte Eingabe" angegeben ist, muss der angegebene Eingabewert vorhanden und der entsprechende Eingangsport angeschlossen sein.

Ist der Parameter "Gewählte Eingabe" nicht angegeben ist, dient der Block als Schutzschalter.

Sie können die ursprünglichen Standardnamen für die Eingabe verwenden. Sie können sie jedoch auch umbenennen, um sie aussagekräftiger zu beschreiben. Die Eingabenamen müssen eindeutig sein, d. h. zwei Eingaben können nicht denselben Namen haben. Verbundene Eingaben müssen alle vom gleichen Typ sein.

Der erwartete Anwendungsfall ist, dass der Parameter "Gewählte Eingabe" als Vorlagenparameter gesetzt wird. Dieser kann dann individuell für jede Modellinstanz festgelegt werden, um zu bestimmen, welche Eingabe verwendet wird.

Parameter

Name	Beschreibung	Typ	Hinweise
Gewählte Eingabe	Um eine Ausgabe zu definieren, geben Sie einen der fünf Eingabenamen an. Um als Schutzschalter zu dienen, lassen Sie diesen Parameter leer.	string	Optional
Name für input1	Der Name des ersten Eingabewertes für den Block.	string	Standard: input1
Name für input2	Der Name des zweiten Eingabewertes für den Block.	string	Standard: input2
Name für input3	Der Name des dritten Eingabewertes für den Block.	string	Standard: input3
Name für input4	Der Name des vierten Eingabewertes für den Block.	string	Standard: input4
Name für input5	Der Name des fünften Eingabewertes für den Block.	string	Standard: input5

Details zum Eingangsport

Name	Beschreibung	Typ
input1	Die erste Eingabe.	any
input2	Die zweite Eingabe.	any
input3	Die dritte Eingabe.	any
input4	Die vierte Eingabe.	any
input5	Die fünfte Eingabe	any

Details zum Ausgangsport

Name	Beschreibung	Typ
Schalter	Der festgelegte Eingabewert, oder kein Wert (Schutzschalter), falls der Parameter "Eingabeparameter" leer ist.	sameAsAll(input1, input2, input3, input4, input5)

Selektor

`apama.analyticsbuilder.blocks.Selector`

Gibt einen Parameterwert aus, abhängig davon, welcher Eingangsport einen `true`-Wert hat, wobei die niedrigste Zahl Vorrang hat.

Mit den Parametern dieses Blocks legen Sie den Ausgabewert fest, der gesendet werden soll. Es wird nur einer der Parameterwerte im Ausgang gesendet. Er wird gesendet, wenn der entsprechende Eingangsport einen `true`-Wert empfängt. Wenn mehr als ein Eingangsport einen `true`-Wert empfängt, dann wird der Eingangsport verwendet, der die niedrigste Zahl in seinem Namen hat. Zum Beispiel hat Eingang 1 eine höhere Priorität als Eingang 2.

Wenn alle Eingabewerte `false` sind, dann wird der mit dem Parameter "Keine Eingabe" angegebene Wert gesendet.

Beispiel: Eingabe 1 hat den Wert "hoch", Eingabe 2 "mittel", Eingabe 3 "gering" und der Parameter "Keine Eingabe" hat den Wert "aus". Wenn keiner der Eingangsports einen `true`-Wert empfängt, dann wird "aus" als Ausgabewert gesendet. Wenn sowohl der Eingangsport 2 als auch der Eingangsport 3 einen `true`-Wert erhalten und der Eingangsport 1 einen `false`-Wert erhält, dann wird als Ausgabewert "medium" gesendet. Dies liegt daran, dass Eingabe 2 eine höhere Priorität hat als Eingabe 3.

Parameter

Name	Beschreibung	Typ	Hinweise
Eingabe 1	Die Ausgabe, die gesendet wird, wenn der Eingangsport "Eingabe 1" den Wert `true` erhält.	string	
Eingabe 2	Die Ausgabe, die gesendet wird, wenn der Eingangsport "Eingabe 2" den Wert `true` erhält, jedoch nur, wenn dies der Port mit der niedrigsten Zahl im Namen ist.	string	Optional
Eingabe 3	Die Ausgabe, die gesendet wird, wenn der Eingangsport "Eingabe 3" den Wert `true` erhält, jedoch nur, wenn dies der Port mit der niedrigsten Zahl im Namen ist.	string	Optional
Eingabe 4	Die Ausgabe, die gesendet wird, wenn der Eingangsport "Eingabe 4" den Wert `true` erhält, jedoch nur, wenn dies der Port mit der niedrigsten Zahl im Namen ist.	string	Optional
Eingabe 5	Die Ausgabe, die gesendet wird, wenn der Eingangsport "Eingabe 5" den Wert `true` erhält, jedoch nur, wenn dies der Port mit der niedrigsten Zahl im Namen ist.	string	Optional

Name	Beschreibung	Typ	Hinweise
Keine Eingabe	Die Ausgabe, die gesendet wird, wenn keine Eingabe den Wert `true` hat.	string	
Typ	Der für den Ausgabewert zu verwendende Typ.	Option - eine von: ■ Float ■ Boolean ■ String	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Eingabe 1	Verursacht, dass der Parameterwert "Eingabe 1" gesendet wird, wenn `true`.	boolean
Eingabe 2	Verursacht, dass der Parameterwert "Eingabe 2" gesendet wird, wenn `true` (und falls keine Eingabe mit niedrigerer Zahl `true` ist).	boolean
Eingabe 3	Verursacht, dass der Parameterwert "Eingabe 3" gesendet wird, wenn `true` (und falls keine Eingabe mit niedrigerer Zahl `true` ist).	boolean
Eingabe 4	Verursacht, dass der Parameterwert "Eingabe 4" gesendet wird, wenn `true` (und wenn keine Eingabe mit niedrigerer Zahl `true` ist).	boolean
Eingabe 5	Verursacht, dass der Parameterwert "Eingabe 5" gesendet wird, wenn `true` (und wenn keine Eingabe mit niedrigerer Zahl `true` ist).	boolean

Details zum Ausgangsport

Name	Beschreibung	Typ
Ausgabe	Der Ausgabewert von einem der Parameter.	any

Zeitverzögerung

`apama.analyticskit.blocks.core.TimeDelay`

Verzögert die Eingabe um die angegebene Zeit.

Parameter

Name	Beschreibung	Typ	Hinweise
Verzögerung (Sek.)	Die Zeitdauer (in Sekunden), um die die Eingabe verzögert wird. Dies muss eine endliche und positive Zahl sein.	float	

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der zu verzögernde Eingabewert.	any

Details zum Ausgangsport

Name	Beschreibung	Typ
Verzögerter Wert	Der verzögerte Ausgabewert.	input(value)

Werkzeug

Cron Timer

`apama.analyticskit.blocks.core.CronTimer`

Sendet eine Signalausgabe basierend auf einer Cron-ähnlichen periodischen Timer-Syntax.

Die Ausgabe "Timer Tick" wird jedes Mal gesendet, wenn die Zeit mit dem Muster übereinstimmt. Jeder Parameter kann auf ein Sternchen (*) gesetzt werden, um alle Zeiten abzudecken, auf eine Liste mit durch Komma getrennten Zahlen, um zu diesen bestimmten Zeiten auszulösen, oder auf "* / Zahl", um bei jedem Vielfachen der Zahl auszulösen.

Zum Beispiel:

Um jeden Mittwoch um 20:30 Uhr ein Signal zu senden:

- Tag der Woche: 4
- Stunde: 20
- Minute: 30

Um ein Signal alle fünf Sekunden zu senden:

- Sekunden: * / 5

Die Parameter "Monatstage" und "Wochentage" wirken so zusammen, dass jede Übereinstimmung in einem der beiden Parameter eine Ausgabe auslöst, wenn beide angegeben sind. Wenn z. B. "Monatstage" auf 15 und "Wochentage" auf 1 eingestellt ist, erfolgt eine Ausgabe an jedem Montag des Monats und am 15., unabhängig davon, welcher Tag das ist.

Dieser Block wird im Simulationsmodus nicht unterstützt.

Parameter

Name	Beschreibung	Typ	Hinweise
Sekunden	Die Sekunden, in denen das Signal ausgelöst wird. Bereich: 0 bis 59.	string	Standard: 0
Minuten	Die Minuten, in denen das Signal ausgelöst wird. Bereich: 0 bis 59.	string	Standard: *
Stunden	Die Stunden, in denen das Signal ausgelöst wird. Bereich: 0 bis 23.	string	Standard: *
Monatstage	Die Monatstage, an denen das Signal ausgelöst wird. Bereich: 1 bis 31.	string	Standard: *
Monate	Die Monate, in denen das Signal ausgelöst wird. Bereich: 1 bis 12.	string	Standard: *
Wochentage	Die Wochentage, an denen das Signal ausgelöst wird. Bereich: 0 (Sonntag) bis 6 (Samstag).	string	Standard: *

Details zum Ausgangsport

Name	Beschreibung	Typ
Timer Tick	Wird zu jeder Auslösezeit entsprechend der Cron-Parameter gesendet.	pulse

Dauer

`apama.analyticskit.blocks.core.Duration`

Misst die Zeit, die seit einer festgelegten Startzeit vergangen ist.

Die Startzeit wird durch ein Startsignal festgelegt, das den Block aktiviert. Wenn der Block bereits aktiv ist, wird das Startsignal ignoriert und die bestehende Messung bleibt unbeeinflusst. Der Block wird mit einem Reset-Signal deaktiviert, was auch regelmäßig wiederkehrende Ausgaben deaktiviert.

Der Block erzeugt eine Float-Ausgabe der seit dem Startsignal verstrichenen Zeit, gemessen in Sekunden. Wenn der Block zum Zeitpunkt des Empfangs eines Messsignals inaktiv ist, wird 0.0 als Ausgabe generiert.

Wenn mehrere Signale gleichzeitig empfangen werden, werden sie in der Reihenfolge "Messung", "Zurücksetzen" und "Start" verarbeitet. Wenn also zum Beispiel die Signale "Messung" und "Zurücksetzen" zusammen empfangen werden, erzeugt der Block zunächst eine Ausgabe mit der aktuellen Dauer und setzt dann seinen Status zurück. Die Verarbeitung der Eingaben in dieser Reihenfolge ermöglicht die Ausgabe des aktuellen Status des Blocks und den Neustart des Blocks, falls gewünscht, innerhalb einer einzigen Zeiteinheit.

Parameter

Name	Beschreibung	Typ	Hinweise
Regelmäßig wiederkehrende Ausgabe (sek)	Die Zeitdauer (in Sekunden) zwischen automatisch generierten Ausgaben durch einen aktiven Block. Wenn nicht definiert, wird eine Ausgabe nur erzeugt, wenn ein Messsignal empfangen wird. Dies muss eine endliche und positive Zahl sein.	float	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Messung	Löst eine Ausgabe der Dauer (in Sekunden) seit Aktivierung des Blocks aus. Wenn der Block inaktiv ist, wird 0.0 ausgegeben.	pulse
Zurücksetzen	Deaktiviert und setzt den Status des Blocks zurück.	pulse
Start	Aktiviert den Block.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Dauer	Die Zeit in Sekunden seit dem Empfang eines Startsignals.	float

Eigenschaften festlegen

`apama.analyticsbuilder.blocks.SetProperties`

Gibt einen Pulse mit Eigenschaften aus, die aus Werten an den Eingangsports festgelegt wurden.

Die Eigenschaftsnamen werden aus den Parametern und die Werte aus den Eingangsports übernommen. Neue Eigenschaften werden nur ausgegeben, wenn sie empfangen wurden. Eine Ausgabe wird gesendet, sobald mindestens eine der Eingaben vorhanden ist.

Eigenschaften werden in der folgenden Rangfolge am Ausgang festgelegt:

1. Alle Eigenschaften, die explizit durch die Verwendung eines Parameters angegeben wurden. Wenn eine Eingabe ein Pulse ist, wird sie als ein Objekt behandelt, das die Eigenschaften dieser Eingabe verwendet. Wenn eine Eingabe einen Primärwert (keinen Pulse) hat, dann wird der Primärwert verwendet. Um stattdessen die Eigenschaften zu verwenden, verwenden Sie den Block "Eigenschaft extrahieren", wobei der Parameter "Eigenschaftspfad" nicht festgelegt ist und der Parameter "Eigenschaftstyp" auf "Eigenschaften" gesetzt ist, wodurch der Primärwert durch einen Pulse (der ignoriert wird) ersetzt wird und die Eigenschaften verwendet werden.
2. Die Eigenschaften eines beliebigen `Value`-Objekts an einem Eingangsport, das nicht über den entsprechenden Parametersatz verfügt. Dies ist ein direkter Merge des Dictionary `properties`. Wenn zwei Dictionaries denselben Eigenschaftsschlüssel haben, dann hat der Eingangsport mit der niedrigsten Identifizierer Vorrang. Somit überschreiben alle gemeinsam genutzten Eigenschaften an Eingang 1 die Eigenschaften von Eingang 2 und abwärts.
3. Alle Eigenschaften eines `Value`-Objekts, die dem Eingangsport "Merge" zur Verfügung gestellt werden, bleiben erhalten, wenn sie nicht durch eine der obigen Vorgänge überschrieben werden. Der optionale Eingangsport "Merge" ermöglicht das Verketten oder Ergänzen eines Satzes von Eigenschaften aus einem anderen Block.

Daher werden alle Eigenschaften, die bei einem Eingang festgelegt sind, von denen mit demselben Namen an einem Eingang mit höherer Priorität überschrieben, oder wenn ein Eingang für die angegebene Eigenschaft konfiguriert ist.

Parameter

Name	Beschreibung	Typ	Hinweise
Eigenschaft 1	Eine Eigenschaft, die in der Ausgabe anhand des Wertes von Eingangsport 1 gesetzt wird.	string	Optional
Eigenschaft 2	Eine Eigenschaft, die in der Ausgabe anhand des Werts von Eingangsport 2 gesetzt wird.	string	Optional
Eigenschaft 3	Eine Eigenschaft, die in der Ausgabe anhand des Wertes von Eingangsport 3 gesetzt wird.	string	Optional
Eigenschaft 4	Eine Eigenschaft, die in der Ausgabe anhand des Wertes von Eingangsport 4 gesetzt wird.	string	Optional

Name	Beschreibung	Typ	Hinweise
Eigenschaft 5	Eine Eigenschaft, die in der Ausgabe anhand des Wertes von Eingangsport 5 gesetzt wird.	string	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Eingabe 1	Wert, der anhand des Eigenschaftsnamen im Parameter "Eigenschaft 1" hinzugefügt werden soll.	any
Eingabe 2	Wert, der anhand des Eigenschaftsnamen im Parameter "Eigenschaft 2" hinzugefügt werden soll.	any
Eingabe 3	Wert, der anhand des Eigenschaftsnamen im Parameter "Eigenschaft 3" hinzugefügt werden soll.	any
Eingabe 4	Wert, der anhand des Eigenschaftsnamen im Parameter "Eigenschaft 4" hinzugefügt werden soll.	any
Eingabe 5	Wert, der anhand des Eigenschaftsnamen im Parameter "Eigenschaft 5" hinzugefügt werden soll.	any
Merge	Quelle, die mit den angegebenen Eigenschaften zusammengeführt wird. Eigenschaften von hier, die nicht durch einen Eingabewert ersetzt werden, werden an den Ausgang weitergeleitet.	any

Details zum Ausgangsport

Name	Beschreibung	Typ
Ausgabe	Der Ausgabewert mit zusätzlichen Eigenschaften.	pulse

Eigenschaft extrahieren

`apama.analyticskit.blocks.core.ExtractProperty`

Extrahiert die angegebene Eigenschaft aus dem Eingabewert und konvertiert sie in den angegebenen Typ.

Der Wert im Wert-Port, der vom Parameter "Eigenschaftspfad" bestimmt wird, sollte eine Zeichenfolge, eine Zahl oder ein boolescher Wert sein.

Sie können einen Punkt (.) als Teil des Parameters "Eigenschaftspfad" angeben, um verschachtelte Werte aus einem Dictionary zu extrahieren.

Beispiel: Wenn die Eingabe `{ "location" : { "city" : "Cambridge" } }` (in JSON-Form) ist, können Sie diesen Wert extrahieren, indem Sie `location.city` als Parameter "Eigenschaftspfad" angeben.

Der Parameter Eigenschaftspfad kann zudem eckige Klammern enthalten, um ein bestimmtes Element aus einer Sequenz zu extrahieren.

Wenn der Wert ein Objekt ist, werden die Eigenschaften dieses Objekts als Eigenschaften am Ausgangsport Extrahierter Wert ausgegeben.

Beispiel: Wenn die Eingabe`{ "users" : [ { "age" : 40.375 } ] }` (in JSON-Form) ist, können Sie diesen Wert extrahieren, indem Sie als Parameter "Eigenschaftspfad" `users[0].age` angeben.

Der Block unterstützt keine Extraktion von Einträgen aus einem Dictionary, dessen Schlüssel Sonderzeichen wie Punkt (.) oder eckige Klammern enthält. Außerdem wird das Extrahieren von Einträgen aus einer Sequenz ohne Verwendung eckiger Klammern nicht unterstützt. Beispielsweise muss `users.0.id` als `users [0] .id` geschrieben werden.

Bei der Umwandlung eines Strings in einen Float behandelt dieser Block einen leeren String als einen Wert von 0,0 statt als nicht analysierbar.

Parameter

Name	Beschreibung	Typ	Hinweise
Eigenschafts-pfad	Der Name oder Pfad der Eigenschaft, die aus dem Eingabewert extrahiert werden soll. Wenn nicht gesetzt, werden alle Eigenschaften ausgegeben.	string	Optional
Eigenschaftstyp	Der Typ, in den der Eigenschaftswert konvertiert werden soll. Wenn auf "Eigenschaften" gesetzt, wird ein Pulse mit Eigenschaften aus dem extrahierten Wert ausgegeben.	Option - eine von: ■ String ■ Boolean ■ Float ■ Eigen-schaften	Standard: String
Beim Fehlen löschen	Wenn gewählt, wird der Standardwert des angegebenen Typs ausgegeben, wenn der Parameter "Eigenschaftspfad" nicht angegeben wurde oder wenn der Wert nicht in den angegebenen Typ umgewandelt werden kann.	boolean	Standard: false

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Der Eingabewert, aus dem die Eigenschaft extrahiert werden soll.	any

Details zum Ausgangsport

Name	Beschreibung	Typ
Extrahierter Wert	Der aus dem Eingabewert extrahierte Wert.	any

Fehlende Daten

`apama.analyticskit.blocks.core.MissingData`

Erzeugt eine Ausgabe, wenn innerhalb einer bestimmten Zeit keine Eingabe erfolgt ist.

Optional kann dieser Block die Abwesenheit von Änderungen erkennen, wenn wiederholte Eingaben desselben Wertes nicht gezählt werden.

Parameter

Name	Beschreibung	Typ	Hinweise
Wiederholte Eingaben ignorieren	Wenn gewählt, werden Eingaben mit demselben Wert ignoriert.	boolean	Standard: false
Dauer (Sek.)	Die Zeitdauer (in Sekunden), die der Block auf eine Eingabe wartet. Dies muss eine endliche und positive Zahl sein.	float	

Details zum Eingangsport

Name	Beschreibung	Typ
Wert	Die Daten, deren Vorhandensein oder Abwesenheit erkannt wird.	any
Zurücksetzen	Setzt den Zustand des Blocks zurück - Timer und der vorherige Wert werden zurückgesetzt.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Fehlende Daten	Abwesenheit von Daten; `true`, wenn innerhalb der angegebenen Zeitdauer keine Eingabe erhalten wird, andernfalls `false`.	boolean

Geofence

`apama.analyticsbuilder.blocks.Geofence`

Vergleicht den Eingabewert mit dem definierten Geofence-Wert, um zu erkennen, ob sich das Gerät innerhalb des Geofence befindet, und ob das Gerät den Geofence betreten oder verlassen hat.

Die Grenzlinie selbst wird als ausserhalb des Geofence-Gebietes betrachtet.

Parameter

Name	Beschreibung	Typ	Hinweise
Geofence	Ein Polygon, das ein Gebiet auf der Erdoberfläche beschreibt. Hinweis: Das Überschreiten der internationalen Datumsgrenze, also des Längengrades von +/- 180 Grad, wird nicht unterstützt.	sequence	

Details zum Eingangsport

Name	Beschreibung	Typ
Position	Der Eingabewert, aus dem die Eigenschaften für den Längengrad (lng) und den Breitengrad (lat) extrahiert werden.	any

Details zum Ausgangsport

Name	Beschreibung	Typ
Innerhalb Geofence	Wird auf `true` gesetzt, wenn sich das Gerät innerhalb des definierten Geofence befindet.	boolean
Geofence betreten	Sendet ein Signal, wenn das Gerät den definierten Geofence betritt.	pulse

Name	Beschreibung	Typ
Geofence verlassen	Sendet ein Signal, wenn das Gerät den definierten Geofence verlässt.	pulse

Konstanter Wert

`apama.analyticsbuilder.blocks.ConstantValue`

Gibt einen Wert aus, entweder wenn der Eingangsport "Trigger" ein Signal empfängt oder beim Start.

Der Eingangsport "Trigger" kann verwendet werden, um die Ausgabe zu verzögern, bis ein Trigger-Eingang empfangen wird. Wenn der Eingangsport "Trigger" nicht angeschlossen ist, gibt der Block bei Aktivierung des Modells einen Wert aus, der eine weitere Verarbeitung auslösen könnte.

Der Parameter "Wert" kann entweder als Zeichenfolge oder als JSON-Wert behandelt werden. Für den JSON-Strings Zahl oder Boolean, ist dies dann der Typ der Ausgabe. Bei JSON-Objekten ist die Ausgabe ein Pulse mit den Eigenschaften des Objekts. JSON-Arrays sind nur innerhalb eines Objekts zulässig.

Parameter

Name	Beschreibung	Typ	Hinweise
Wert	Der auszugebende Wert.	string	
Typ	So wird der Parameter "Wert" interpretiert.	Option - eine von: ■ string ■ JSON	Standard: JSON

Details zum Eingangsport

Name	Beschreibung	Typ
Trigger	Wenn verbunden, Ausgabe an diesen Trigger senden.	boolean

Details zum Ausgangsport

Name	Beschreibung	Typ
Ausgabe	Der Ausgabewert des Parameters "Wert".	any

Textersetzung

`apama.analyticsbuilder.blocks.TextSubstitution`

Ersetzt Identifizierer, die mit einem Hash und geschweiften Klammern gekennzeichnet sind (zum Beispiel, `#{name}`) in der Textvorlage durch entsprechende Eingaben aus den Eingabewerten.

Mindestens einer der Eingangsports "Objekt" oder "Quelle" muss verbunden sein. Identifizierer, die nicht aufgelöst werden können, werden nicht ersetzt.

Die Identifizierer mit dem Präfix `source.` (zum Beispiel `#{source.name}`) werden in dem am Eingangsport "Quelle" empfangenen Wert gesucht. Wenn der am Eingangsport "Quelle" empfangene Wert zum Beispiel `{ "name": "sample_name" }` ist, dann wird der Identifizierer `#{source.name}` aufgelöst in `sample_name`.

Die Identifizierer, denen kein `source.` vorangestellt ist, werden in dem am Eingangsport "Objekt" empfangenen Wert gesucht.

Eingebettete Identifizierer können angegeben werden, indem sie durch einen Punkt (.) getrennt werden. Zum Beispiel, wenn der Eingangsport "Objekt" den Wert `{ "address": { "street": { "name": "example_street" }}}` hat, dann wird der Identifizierer `#{address.street.name}` aufgelöst in `example_street`.

Schlüssel, die einen Punkt (.) enthalten, werden nicht unterstützt. Wenn etwa der Wert des Eingangsports "Objekt" die Form `{ "address.street": { "name": "example_street" }}` hat, dann kann der Wert `example_street` nicht aufgelöst werden, weil der Identifizierer `#{address.street.name}` erwartet, dass `street` im Eintrag `address` eingebettet ist.

Primitive Werte wie `integer`, `float`, `boolean` und `string` werden direkt ersetzt, aber komplexe Werte werden vor der Ersetzung in eine JSON-Darstellung konvertiert.

Jeder Identifizierer, der den Text `time` (Groß-/Kleinschreibung wird nicht beachtet) und den Wertetyp `float` enthält, wird als Zeitstempelwert interpretiert und vor der Ersetzung in das Format `yyyy-MM-ddTHH:mm:ss.SSSZ` umgewandelt. Dies kann mit optionalen Parametern anhand der folgenden Syntax geändert werden: `{time:param1="value1",param2="value2"}`. Verwenden Sie den Parameter `TZ="time_zone"`, um eine andere Zeitzone anzugeben und/oder verwenden Sie den Parameter `FORMAT="format_string"`, um ein anderes Format anzugeben. So gibt zum Beispiel `#{time:TZ="America/New_York",FORMAT="HH:mm:ssZ"}` die Zeitzone von New York an und das Zeitformat ist `HH:mm:ssZ`. Das Modell lässt sich nicht aktivieren, wenn die Zeitzone nicht erkannt wird oder das Format ungültig ist.

Hinweis: Die Zeichenfolge für das Zeitformat darf weder Anführungszeichen (") noch geschweifte Klammern ({ und }) enthalten.

Ein Hash (#) kann in der Textvorlage angegeben werden, indem man es wie folgt escaped: `#{#}`.

Weitere Informationen finden Sie in der Liste der Zeitzonen und der Liste der gültigen Zeichenfolgen für das Zeitformat in der Apama-Dokumentation.

Parameter

Name	Beschreibung	Typ	Hinweise
Textvorlage	Der Text, der zum Generieren der Ausgabe verwendet wird, indem die darin enthaltenen Identifizierer wie #{name} durch die Werte der Eingangports ersetzt werden.	string	

Details zum Eingangsport

Name	Beschreibung	Typ
Objekt	Wird verwendet, um Identifizierer zu ersetzen, denen nicht das Präfix `source.` vorangestellt ist. Zum Beispiel: `#{name}` or `#{timestamp}`.	any
Quelle	Wird verwendet, um Identifizierer zu ersetzen, denen das Präfix `source.` vorangestellt ist. Zum Beispiel: `#{source.name}`.	any

Details zum Ausgangsport

Name	Beschreibung	Typ
Ausgabe	String, der die Ersetzungen aus der Textvorlage enthält.	string

Umschalten

`apama.analyticskit.blocks.core.Toggle`

Konvertiert zwei Pulse-Eingaben in eine Boolean-Ausgabe auf Grundlage der Pulse-zum Setzen und Zurücksetzen. Verzögerungen können optional eingerichtet werden.

Ohne Verzögerungen wird der Ausgabestatus bei einem Setzen-Signal zu `true` geändert und bei einem Zurücksetzen-Pulse zu `false`. Wenn beide Pulse gleichzeitig empfangen werden, wird der Ausgabestatus umgeschaltet (zu `true`, wenn Signale zum ersten Mal erhalten werden).

Wenn Verzögerungszeiten angegeben sind, wird der Ausgabestatus erst nach Anwendung der Verzögerung zu `true` oder `false` geändert

Die folgende Ausnahme gilt, wenn beide Pulse gleichzeitig empfangen werden: Der Ausgabestatus wird nur umgeschaltet, wenn beide Verzögerungszeiten gleich sind oder nicht angegeben wurden.

Parameter

Name	Beschreibung	Typ	Hinweise
Verzögerung festlegen (Sek.)	Die Zeitdauer (in Sekunden), nach der der gesetzte Pulse verarbeitet wird. Wenn der Parameter nicht angegeben wird, dann wird der Pulse sofort verarbeitet. Dies muss eine endliche und positive Zahl sein.	float	Optional
Verzögerung bei Zurücksetzen (Sek.)	Die Zeitdauer (in Sekunden), nach der ein Pulse zum Zurücksetzen verarbeitet wird. Wenn der Parameter nicht angegeben ist, dann wird der Pulse sofort verarbeitet. Dies muss eine endliche und positive Zahl sein.	float	Optional

Details zum Eingangsport

Name	Beschreibung	Typ
Setzen	Setzt die Ausgabe auf `true`. Ausstehende verzögerte Pulse zum Setzen oder Zurücksetzen werden mit der neuen Ausgabe storniert.	pulse
Zurücksetzen	Setzt die Ausgabe auf `false`. Ausstehende verzögerte Pulse zum Setzen oder Zurücksetzen werden mit der neuen Eingabe storniert.	pulse

Details zum Ausgangsport

Name	Beschreibung	Typ
Wert	Die aus den Eingabe-Pulsen erzeugten Ausgabewerte.	boolean