



# ibaPDA-Data-Store-Kafka

## Daten Streaming zu Apache Kafka-Cluster

Handbuch  
Ausgabe 1.4

Messsysteme für Industrie und Energie  
[www.iba-ag.com](http://www.iba-ag.com)

---

## Hersteller

iba AG  
Königswarterstraße 44  
90762 Fürth  
Deutschland

## Kontakte

Zentrale	+49 911 97282-0
Support	+49 911 97282-14
Technik	+49 911 97282-13
E-Mail	iba@iba-ag.com
Web	www.iba-ag.com

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts sind nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz.

© iba AG 2025, alle Rechte vorbehalten.

Der Inhalt dieser Druckschrift wurde auf Übereinstimmung mit der beschriebenen Hard- und Software überprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass für die vollständige Übereinstimmung keine Garantie übernommen werden kann. Die Angaben in dieser Druckschrift werden jedoch regelmäßig aktualisiert. Notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten oder können über das Internet heruntergeladen werden.

Die aktuelle Version liegt auf unserer Website [www.iba-ag.com](http://www.iba-ag.com) zum Download bereit.

Version	Datum	Revision	Autor	Version SW
1.4	03-2025	Änderungen bei Event Hub	st, mm	8.10.0

Windows® ist eine Marke und eingetragenes Warenzeichen der Microsoft Corporation. Andere in diesem Handbuch erwähnte Produkt- und Firmennamen können Marken oder Handelsnamen der jeweiligen Eigentümer sein.

## Inhalt

<b>1</b>	<b>Zu dieser Dokumentation .....</b>	<b>4</b>
1.1	Zielgruppe und Vorkenntnisse .....	4
1.2	Schreibweisen.....	5
1.3	Verwendete Symbole.....	6
<b>2</b>	<b>Einleitung.....</b>	<b>7</b>
2.1	Systemvoraussetzungen .....	8
<b>3</b>	<b>Konfiguration der Datenaufzeichnung .....</b>	<b>9</b>
3.1	Datenaufzeichnung hinzufügen .....	9
3.2	Datenaufzeichnung Kafka-Cluster.....	10
3.2.1	Typ Kafka.....	11
3.2.2	Typ Event Hub.....	16
3.3	Puffer .....	17
3.4	Konfiguration der Topics .....	20
<b>4</b>	<b>Signalauswahl .....</b>	<b>23</b>
<b>5</b>	<b>Trigger-Einstellung.....</b>	<b>25</b>
<b>6</b>	<b>Diagnose .....</b>	<b>29</b>
6.1	Status der Datenaufzeichnung.....	29
6.2	Diagnose der Datenaufzeichnung.....	30
<b>7</b>	<b>Anhang .....</b>	<b>31</b>
7.1	Beschreibung der Datenformate .....	31
7.1.1	Datenformat JSON (gruppiert).....	31
7.1.2	Datenformat JSON (pro Signal) .....	33
7.1.3	Datenformat AVRO (pro Signal) .....	35
<b>8</b>	<b>Support und Kontakt.....</b>	<b>37</b>

# 1 Zu dieser Dokumentation

Diese Dokumentation beschreibt die Funktion und Anwendung der Datenaufzeichnung *ibaPDA-Data-Store-Kafka*.

Diese Dokumentation ist eine Ergänzung zum *ibaPDA*-Handbuch. Informationen über alle weiteren Eigenschaften und Funktionen von *ibaPDA* finden Sie im *ibaPDA*-Handbuch bzw. in der Online-Hilfe.

Grundlegende Informationen zur Datenaufzeichnung in *ibaPDA* finden Sie im *ibaPDA*-Handbuch Teil 5.

## 1.1 Zielgruppe und Vorkenntnisse

Diese Dokumentation wendet sich an ausgebildete Fachkräfte, die mit dem Umgang mit elektrischen und elektronischen Baugruppen sowie der Kommunikations- und Messtechnik vertraut sind. Als Fachkraft gilt, wer auf Grund der fachlichen Ausbildung, Kenntnisse und Erfahrungen sowie Kenntnis der einschlägigen Bestimmungen die übertragenen Arbeiten beurteilen und mögliche Gefahren erkennen kann.

Im Besonderen wendet sich diese Dokumentation an Personen, die mit Projektierung, Test, Inbetriebnahme oder Instandhaltung der unterstützten Datenbanken, Cloud- bzw. Cluster-Storage-Technologie befasst sind. Für den Umgang mit *ibaPDA-Data-Store-Kafka* sind folgende Vorkenntnisse erforderlich bzw. hilfreich:

- Betriebssystem Windows
- Grundkenntnisse *ibaPDA*
- Grundkenntnisse Datenbanken, Cloud- bzw. Cluster-Storage-Technologie

## 1.2 Schreibweisen

In dieser Dokumentation werden folgende Schreibweisen verwendet:

Aktion	Schreibweise
Menübefehle	Menü <i>Funktionsplan</i>
Aufruf von Menübefehlen	<i>Schritt 1 – Schritt 2 – Schritt 3 – Schritt x</i> Beispiel: Wählen Sie Menü <i>Funktionsplan – Hinzufügen – Neuer Funktionsblock</i>
Tastaturtasten	<Tastename> Beispiel: <Alt>; <F1>
Tastaturtasten gleichzeitig drücken	<Tastename> + <Tastename> Beispiel: <Alt> + <Strg>
Grafische Tasten (Buttons)	<Tastename> Beispiel: <OK>; <Abbrechen>
Dateinamen, Pfade	<i>Dateiname, Pfad</i> Beispiel: <i>Test.docx</i>

## 1.3 Verwendete Symbole

Wenn in dieser Dokumentation Sicherheitshinweise oder andere Hinweise verwendet werden, dann bedeuten diese:

---

### Gefahr!



**Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die unmittelbare Gefahr des Todes oder der schweren Körperverletzung!**

- Beachten Sie die angegebenen Maßnahmen.

---

### Warnung!



**Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die mögliche Gefahr des Todes oder schwerer Körperverletzung!**

- Beachten Sie die angegebenen Maßnahmen.

---

### Vorsicht!



**Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die mögliche Gefahr der Körperverletzung oder des Sachschadens!**

- Beachten Sie die angegebenen Maßnahmen.

---

### Hinweis



Hinweis, wenn es etwas Besonderes zu beachten gibt, wie z. B. Ausnahmen von der Regel usw.

---

### Tipp



Tipp oder Beispiel als hilfreicher Hinweis oder Griff in die Trickkiste, um sich die Arbeit ein wenig zu erleichtern.

---

### Andere Dokumentation



Verweis auf ergänzende Dokumentation oder weiterführende Literatur.

## 2 Einleitung

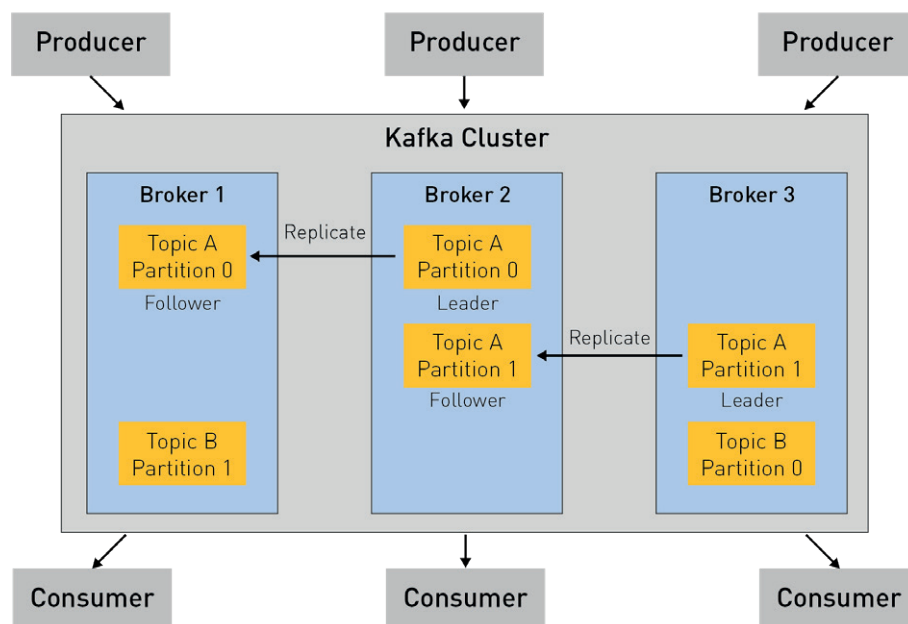
Für unterschiedliche Zwecke und Methoden der Datenspeicherung stehen verschiedene Arten der Datenaufzeichnung in *ibaPDA* zur Verfügung. Je nachdem, welche Lizenzen im Lizenzcontainer freigeschaltet sind, werden im Dialog verschiedene Arten der Datenaufzeichnung zur Konfiguration angeboten.

Die vorliegende Dokumentation beschreibt den Aufzeichnungstyp "Kafka-Cluster zeitbasierte Datenaufzeichnung". Dieser Aufzeichnungstyp schreibt zeitbasierte Daten in einen Kafka-Cluster.

Die Architektur von Apache Kafka besteht aus einem Cluster-Rechnerverbund. In diesem Rechnerverbund speichern sogenannte Broker empfangene Nachrichten mit einem Zeitstempel in sogenannten Topics.

Topics können in mehrere im Cluster verteilte Partitionen aufgeteilt und repliziert werden. Als Producer sind Anwendungen, die Nachrichten in einen Kafka-Cluster schreiben. Dabei werden die Nachrichten an den Leader gesendet, die Replikation mit den Followern erledigen die Broker untereinander. Consumer sind Anwendungen, die Nachrichten vom Kafka-Cluster lesen.

*ibaPDA* hat in einer Kafka-Umgebung die Rolle eines Producers: Die Datenaufzeichnung sendet Nachrichten an den Cluster.



Weiterführende Informationen zu Apache Kafka finden Sie unter

- <https://kafka.apache.org/>
- <https://www.confluent.io/what-is-apache-kafka>

Die Auswahl der Signale, die in den Topics aufgezeichnet werden sollen, ist in Kapitel [↗ Signalauswahl](#), Seite 23 beschrieben.

Die Daten können kontinuierlich oder getriggert aufgezeichnet werden, siehe Kapitel [↗ Trigger-Einstellung](#), Seite 25.

## 2.1 Systemvoraussetzungen

Folgende Systemvoraussetzungen sind für die Verwendung der Datenaufzeichnung in einem Apache Kafka-Cluster erforderlich:

- *ibaPDA* v8.0.0 oder höher
- Lizenz für *ibaPDA-Data-Store-Kafka*

Die Lizenzen sind gestaffelt nach der Anzahl der Signale, die in den Kafka-Cluster geschrieben werden sollen. Die Anzahl der genutzten Datenaufzeichnungen ist nicht beschränkt.

Bestell-Nr.	Produktbezeichnung	Beschreibung
30.670160	ibaPDA-Data-Store-Kafka-16	Zusatzlizenz zum Schreiben von Daten in einen Kafka-Cluster, max. 16 Signale
30.670161	ibaPDA-Data-Store-Kafka-64	Zusatzlizenz zum Schreiben von Daten in einen Kafka-Cluster, max. 64 Signale
30.670162	ibaPDA-Data-Store-Kafka-256	Zusatzlizenz zum Schreiben von Daten in einen Kafka-Cluster, max. 256 Signale
30.670163	ibaPDA-Data-Store-Kafka-1024	Zusatzlizenz zum Schreiben von Daten in einen Kafka-Cluster, max. 1024 Signale
30.670171	upgrade-ibaPDA-Data-Store-Kafka-16 to 64	Lizenz für Erweiterung von 16 auf 64 Signale
30.670172	upgrade-ibaPDA-Data-Store-Kafka-64 to 256	Lizenz für Erweiterung von 64 auf 256 Signale
30.670173	upgrade-ibaPDA-Data-Store-Kafka-256 to 1024	Lizenz für Erweiterung von 256 auf 1024 Signale


Tab. 1: Verfügbare Lizenzen für die Kafka-Cluster Datenaufzeichnung



## 3 Konfiguration der Datenaufzeichnung

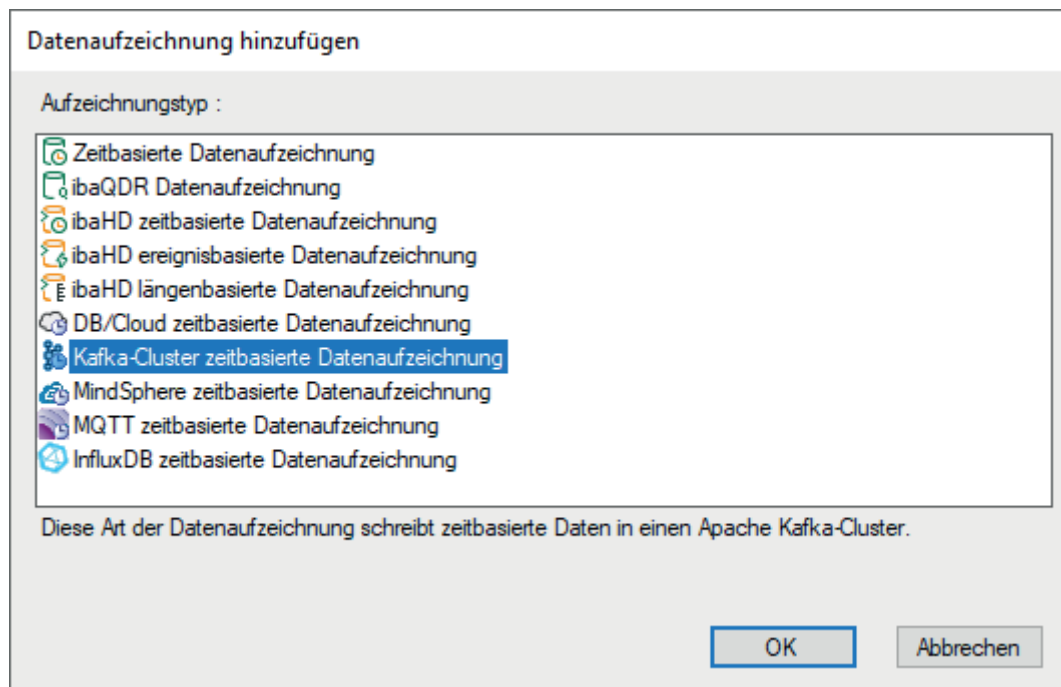
Im Folgenden erfahren Sie, wie Sie eine Datenaufzeichnung konfigurieren.

### 3.1 Datenaufzeichnung hinzufügen

Den Dialog für die Konfiguration der Datenaufzeichnung öffnen Sie über das Hauptmenü *Konfiguration – Datenaufzeichnung* oder mit dem Button  in der Hauptsymbolleiste.

Um eine neue Aufzeichnung hinzuzufügen, klicken Sie auf den blauen Link *Datenaufzeichnung hinzufügen* in der Baumstruktur. Alternativ klicken Sie mit der rechten Maustaste auf einen Aufzeichnungsknoten in der Baumstruktur und wählen Sie *Datenaufzeichnung hinzufügen* aus dem Kontextmenü.

Wählen Sie *Kafka-Cluster zeitbasierte Datenaufzeichnung* für die Aufzeichnung zeitbasierter Daten in einen Apache Kafka-Cluster.



## 3.2 Datenaufzeichnung Kafka-Cluster

Dieses Kapitel beschreibt die Konfiguration der *Kafka-Cluster zeitbasierten Datenaufzeichnung*. Die Konfiguration der Topics nehmen Sie im Unterknoten *Topics* vor, siehe [↗ Konfiguration der Topics](#), Seite 20.

### Allgemein

#### Verriegelt

Sie können eine Datenaufzeichnung verriegeln, um ein versehentliches oder unautorisiertes Ändern der Einstellungen zu verhindern.

#### Aktiv

Aktivieren Sie die Datenaufzeichnung, um Daten aufzuzeichnen. Sie können jedoch verschiedene Aufzeichnungen konfigurieren und nicht benötigte Aufzeichnungen deaktivieren.

### Datenaufzeichnungsindex

Eindeutiger Index aller vorhandenen Kafka-Cluster-Datenaufzeichnungen. Sie müssen diesen Index beispielsweise in der virtuellen Funktion *DataStoreInfoKafka()* referenzieren, um Diagnosedaten für eine bestimmte Kafka-Cluster-Datenaufzeichnung zu erzeugen.

### Name der Datenaufzeichnung

Hier können Sie einen Namen für die Datenaufzeichnung eintragen.

### Identifikator

Der Identifikator ist ein optionaler, textbasierter Wert für die in den Cluster geschriebenen Datensätze. Dieser Wert kann bei der anschließenden Verarbeitung der Daten nützlich sein, um zwischen mehreren *ibaPDA*-Systemen zu unterscheiden, die in denselben Cluster schreiben.

## Verbindung

### Library Version

Anzeige der in *ibaPDA* enthaltenen Library Version (nur zur Information). *ibaPDA* verwendet die Open-Source-Bibliothek *librdkafka*.

### Typ

Zwei Typen stehen zur Auswahl

- Kafka: Generischer Kafka-Cluster, siehe Kapitel [Typ Kafka](#), Seite 11
- Event Hub: für die Verbindung mit einem Microsoft Azure Event Hub mit aktivierter Kafka-Unterstützung, siehe Kapitel [Typ Event Hub](#), Seite 16

## 3.2.1 Typ Kafka

Folgende Einstellungen sind für den Typ **Kafka** erforderlich:

### Cluster-Adresse

Tragen Sie hier Hostname und Port des Brokers im Kafka-Cluster ein. Wenn Sie mehrere Broker gleichzeitig adressieren wollen, trennen Sie die einzelnen Einträge durch Kommas (z. B. 'hostname1:9092, hostname2:9092, hostname3:9092').

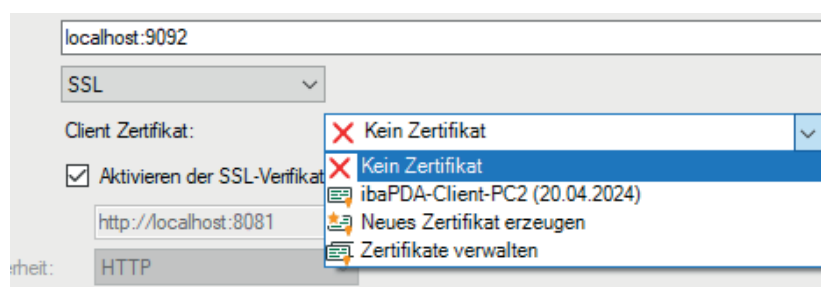
### Cluster-Verbindungssicherheit

Unterschiedliche Verschlüsselungs- und Authentifizierungsmethoden werden unterstützt. Je nach ausgewählter Option sind weitere Eingaben erforderlich.

#### ■ PLAINTEXT

Daten werden im Klartext ausgetauscht, keine weiteren Eingaben

#### ■ SSL

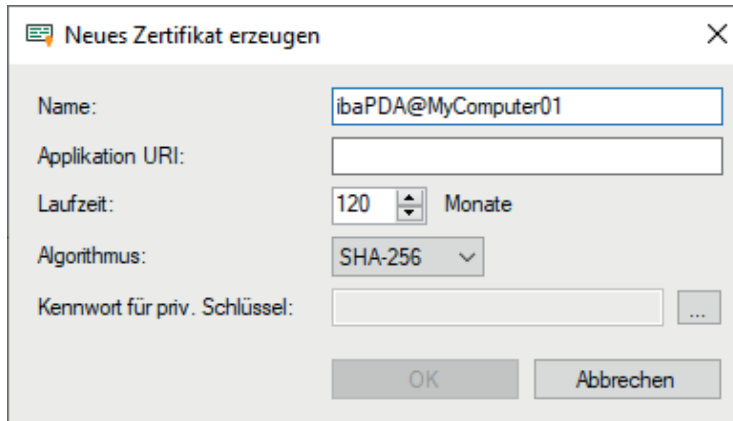


Für die SSL-Kommunikation sind Zertifikate erforderlich. Sie können Zertifikate im zentralen Zertifikatspeicher in *ibaPDA* ablegen und verwalten. Detaillierte Informationen zum Zertifikatspeicher finden Sie im *ibaPDA*-Handbuch.

Im Drop-down-Menü *Client Zertifikat* stehen die verwendbaren Zertifikate zur Auswahl. Darüber hinaus stehen weitere Optionen zur Auswahl.

- *Kein Zertifikat*: Es wird kein Zertifikat verwendet. Dies führt jedoch normalerweise dazu, dass die Validierung fehlschlägt.
- *Zertifikate verwalten*: Öffnet den zentralen Zertifikatspeicher, in dem die Zertifikate zentral in *ibaPDA* verwaltet werden können.

- **Neues Zertifikat erzeugen:** *ibaPDA* erzeugt ein neues selbstsigniertes Zertifikat. Die notwendigen Einstellungen nehmen Sie im sich öffnenden Dialog vor. Nach erfolgreichem Erzeugen des Zertifikats ist das neue Zertifikat ausgewählt.



Geben Sie einen Namen für das Zertifikat ein. Den voreingestellten Namen können Sie ändern.

Die Eingabe einer Application URI ist optional. Stellen Sie die Laufzeit ein und wählen den Algorithmus aus. Zur Auswahl stehen SHA-256, SHA-384 und SHA-512.

Sie müssen ein Kennwort vergeben, um ein Zertifikat zu erzeugen. Um das Kennwort für den privaten Schlüssel einzugeben, klicken Sie auf den Button <...>. Im folgenden Dialog müssen Sie das Kennwort aus Sicherheitsgründen zweimal eingeben. Das Kennwort können Sie beliebig vergeben, es muss keine weiteren Anforderungen erfüllen.

### Aktivieren der SSL-Verifikation

Wenn Sie die SSL-Verifikation aktivieren, wird das Zertifikat des Kafka-Servers automatisch verifiziert. Dazu muss unter den Zertifikaten in *ibaPDA* das Aussteller-Zertifikat vorhanden sein, mit dem das Kafka-Server-Zertifikat signiert wurde.

#### Hinweis



#### Nutzung von verketteten SSL-Zertifikaten

Sie können in *ibaPDA* auch verkettete Zertifikate importieren. Beim Import (z. B. als PEM-Datei oder PFX-Datei) werden die verketteten Zertifikate in einzelne Zertifikate aufgeteilt. Nach dem Import eines verketteten Zertifikats existieren daher mehrere einzelne Einträge unter den Zertifikaten.

Wenn das Kafka-Server-Zertifikat über eine Kette von Aussteller-Zertifikaten signiert wurde und Sie gleichzeitig die SSL-Verifikation nutzen wollen, ist die Konfiguration des Keystores im Kafka-Server wichtig: Wenn der Keystore neben dem Kafka-Server-Zertifikat auch die komplette Kette der Aussteller-Zertifikate enthält, dann muss nur das Zertifikat der Root CA in *ibaPDA* vorhanden sein. Wenn der Keystore dagegen ausschließlich das Kafka-Server-Zertifikat enthält, muss in *ibaPDA* die vollständige Kette der Aussteller-Zertifikate vorhanden sein, damit die Verifizierung möglich ist.

## ■ SASL/PLAINTEXT

Wählen Sie den SASL-Authentifizierungsmechanismus aus:

- PLAIN (alle Daten werden im Klartext ausgetauscht)
- SCRAM-SHA-256
- SCRAM-SHA-512

Geben Sie den Benutzername sowie das Kennwort für die Anmeldung am Kafka-Cluster in den entsprechenden Feldern ein.

## ■ SASL/SSL

Wie bei *SASL/PLAINTEXT* wählen Sie den SASL-Mechanismus aus und geben Benutzername und Kennwort ein. Im Drop-down-Menü *Client Zertifikat* stehen die verwendbaren Zertifikate zur Auswahl.

Für die Aktivierung und Nutzung der SSL-Verifikation gelten ebenfalls die Hinweise aus dem Abschnitt zur Cluster-Verbindungssicherheit mit *SSL*.

## Schema Registry-Adresse

Aktivieren Sie diese Option, wenn im Kafka-Cluster eine Schema Registry genutzt wird. Tragen Sie dann hier Hostname und Port der Schema Registry ein. Je nach genutzter Verbindungssicherheit muss die Adresse mit "http://" oder "https://" beginnen. Standardmäßig ist diese Option deaktiviert.

## Schema Registry-Verbindungssicherheit

*ibaPDA* unterstützt unterschiedliche Methoden für die Verbindungssicherheit. Je nach ausgewählter Option sind weitere Eingaben erforderlich.

## ■ HTTP

Keine weitere Eingabe erforderlich

## ■ HTTPS

Schema-Registry-Verbindungssicherheit: HTTPS

Client Zertifikat: Kein Zertifikat

☒ Aktivieren der SSL-Verifikation

Für die HTTPS-Kommunikation sind Zertifikate erforderlich. Im Drop-down-Menü *Client Zertifikat* stehen die verwendbaren Zertifikate zur Auswahl.

Die Auswahl eines Zertifikats sowie die Aktivierung und Nutzung der SSL-Verifikation erfolgen wie bereits bei der Cluster-Verbindungssicherheit mit SSL beschrieben.

## ■ HTTP + Authentication

Schema-Registry-Verbindungssicherheit: HTTP + Authentication

Benutzername:

Kennwort:  ...

Diese Methode erfordert die Eingabe von Benutzername und Kennwort.

## ■ HTTPS + Authentication

Schema-Registry-Verbindungssicherheit: HTTPS + Authentication

Benutzername:

Kennwort:  ...

Client Zertifikat: Kein Zertifikat

☒ Aktivieren der SSL-Verifikation

Wie HTTPS, zusätzlich ist die Eingabe von Benutzername und Kennwort erforderlich.

## Telegramm-Timeout

Zeit in Sekunden, die *ibaPDA* auf eine Antwort aus dem Kafka-Cluster wartet. Der Wert entspricht dem Parameter *message.timeout.ms* in der Bibliothek *librdkafka*.

## Acknowledgement-Modus

- *Kein*: Datennachrichten werden als kontinuierlicher Strom geendet, jedoch ohne Bestätigung, ob der Leader sie empfangen hat oder nicht. Dies ist der schnellste Modus, aber es ist nicht garantiert, dass der Broker die Daten erfasst hat.
- *Leader*: *ibaPDA* wartet, bis der Topic-Leader die Erfassung der gesendeten Daten bestätigt. Wenn der Leader nach dem Senden der Bestätigung ausfällt und bevor die Follower den Datensatz repliziert haben, gehen die Daten verloren.
- *Alle*: *ibaPDA* sendet keine neuen Daten, bis der Topic-Leader und die Follower die jeweilige Aufzeichnung der Daten bestätigt haben.

Der Wert entspricht dem Parameter *acks* in der Bibliothek *librdkafka*.

## Message Batching Time

Zeit in Millisekunden, die *ibaPDA* wartet, um Nachrichten zu senden. Wenn der Wert 0 ist, werden Nachrichten so schnell wie möglich an den Kafka-Cluster gesendet. Wenn der Wert auf z. B. 100 ms eingestellt ist, werden alle 100 ms Pakete mit gepufferten Nachrichten gesendet. Dies erhöht die Latenzzeit, verringert aber den Verarbeitungsaufwand sowohl auf der Seite von *ibaPDA* als auch auf der Cluster-Seite.

Der Wert entspricht dem Parameter *linger.ms* in der Bibliothek *librdkafka*.

## Zusätzliche Parameter

Sie können zusätzliche Parameter der Bibliothek *librdkafka* konfigurieren.

Zusätzliche Parameter:

Schlüssel	Wert
?	





+

×

↶

↷

In der Tabelle sind bereits angelegte Parameter mit Namen und Wert aufgelistet. Neben der Liste befinden sich Buttons mit folgenden Funktionen:

	Parameter hinzufügen Geben Sie in die Spalten <i>Schlüssel</i> und <i>Wert</i> den Namen und einen Wert ein.
	Markierten Parameter löschen
	Parameter importieren Sie können Parameter aus einer CSV-Datei importieren, indem Sie im Dateibrowser die CSV-Datei auswählen.
	Parameter exportieren Sie können Parameter in eine CSV-Datei exportieren. Geben Sie einen Dateinamen ein und wählen einen Speicherort aus.

## Hinweis



In der Liste konfigurierte Parameter haben immer Vorrang und überschreiben ggf. andere Einstellungen.

Die Dokumentation der in der Bibliothek *librdkafka* verfügbaren Parameter finden Sie unter:

<https://github.com/edenhill/librdkafka/blob/master/CONFIGURATION.md>

### 3.2.2 Typ Event Hub

Folgende Einstellungen sind für den Typ **Event Hub** erforderlich:

The screenshot shows a configuration window titled 'Verbindung'. It contains the following fields and controls:

- Library-Version:** A text box containing '2.3.0'.
- Typ:** A dropdown menu showing 'Event Hub'.
- Verbindung testen:** A button located to the right of the 'Typ' dropdown.
- Endpoint:** An empty text box.
- Key-Name:** An empty text box.
- Key:** An empty text box.
- Entity-Pfad:** An empty text box.
- Connection-String aus Zwischenablage einfügen:** A button located below the 'Entity-Pfad' field.
- Telegramm-Timeout:** A spinner box set to '30' with a unit 's'.
- Aktivieren der SSL-Verifikation:** A checked checkbox.

#### Endpoint, Key-Name, Key, Entity-Pfad

In diese Felder werden die entsprechenden Teile eines Connection-String aus den Eigenschaften des Event Hubs im Microsoft Azure Portal eingefügt. Mit dem Button darunter können Sie die Felder automatisch ausfüllen.

#### <Connection-String aus Zwischenablage einfügen>

Wenn Sie einen Connection-String aus den Eigenschaften des Event Hubs im Microsoft Azure Portal in die Zwischenablage kopiert haben, können Sie ihn mit diesem Button einfügen. Die Felder darüber werden dann automatisch ausgefüllt.

#### Telegramm-Timeout

Zeit in Sekunden, die *ibaPDA* auf eine Antwort aus dem Event Hub wartet.

#### <Verbindung testen>

Mit dem Button <Verbindung testen> können Sie die Verbindung zum Event Hub testen.

#### Aktivieren der SSL-Verifikation

Für die SSL-Verifikation ist das aktuelle CA-Root-Zertifikat "DigiCert Global Root G2" erforderlich, da Azure-Dienste Zertifikate damit signieren. Bei der Installation von *ibaPDA* wird dieses Zertifikat automatisch im Zertifikatsspeicher abgelegt.



### 3.3 Puffer

Die Datenaufzeichnung nutzt einen Speicherpuffer und zusätzlich einen optional aktivierbaren Dateipuffer.

Die Beschreibung gilt für alle Typen von Datenaufzeichnungen, die Daten zu externen Systemen transferieren und bei denen es zu temporären Problemen bei der Erreichbarkeit und verfügbaren Bandbreite kommen kann, wie zum Beispiel:

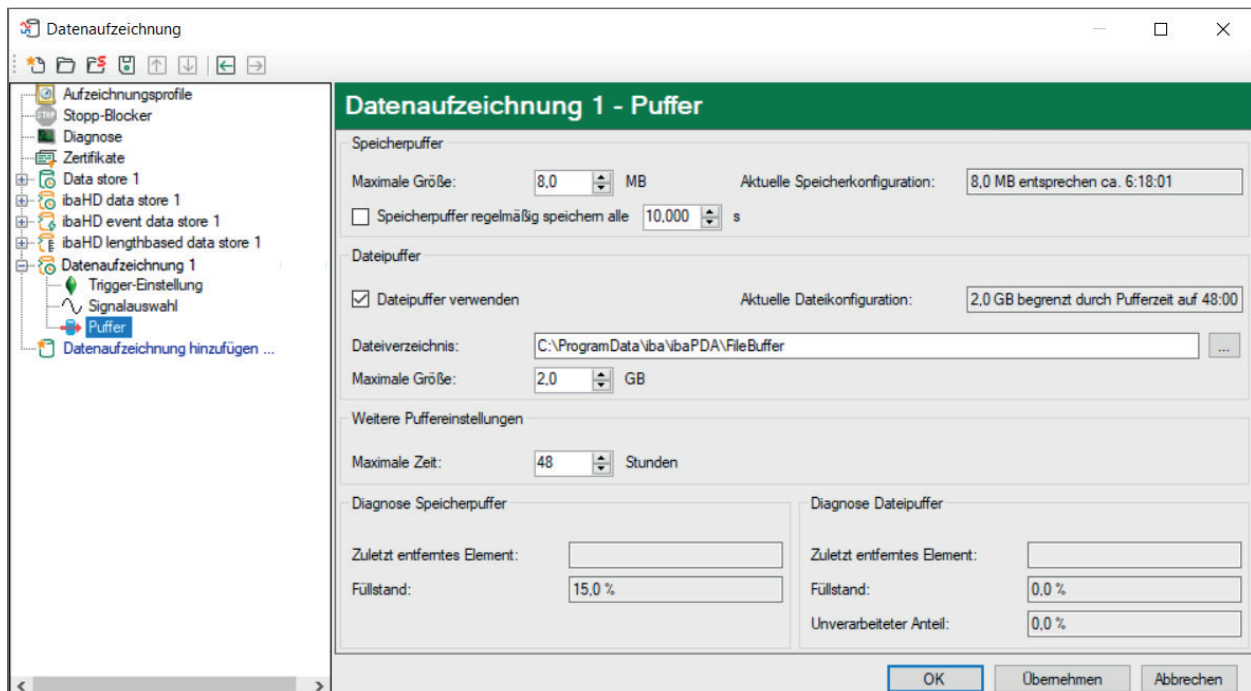
- ibaHD zeitbasiert/ereignisbasiert/längenbasiert
- DB/Cloud zeitbasiert
- Kafka-Cluster zeitbasiert
- MQTT zeitbasiert
- MindSphere zeitbasiert
- InfluxDB zeitbasiert

Daten, die an das Zielsystem gesendet werden sollen, durchlaufen immer den *ibaPDA*-internen Speicherpuffer. Bei bestehender Verbindung zum Zielsystem werden die Daten aus dem Speicherpuffer sofort dorthin gesendet. Geht die Verbindung verloren oder können die Daten nicht schnell genug verschickt werden, dann bleiben die Daten im Speicherpuffer erhalten. Der Speicherpuffer liegt im RAM des *ibaPDA*-Rechners, ist daher limitiert und flüchtig. Wird z. B. die Erfassung neu gestartet, gehen die gepufferten Daten verloren. Wächst der Speicherpuffer bei laufender Erfassung über die konfigurierte Größe hinaus an, werden die ältesten Werte gelöscht und gehen somit verloren.

Zur Verbesserung kann zusätzlich ein Dateipuffer aktiviert werden, der weitaus größere Datenmengen puffern kann. Die Daten werden dabei in Dateien in einem Verzeichnis eines lokalen Laufwerks des *ibaPDA*-Servers gespeichert. Bei aktiviertem Dateipuffer werden Daten aus dem überlaufenden Speicherpuffer in den Dateipuffer übertragen. Wird die Erfassung beendet oder neu gestartet (z. B. durch Anwenden einer geänderten IO-Konfiguration), werden Daten, die sich evtl. zu diesem Zeitpunkt im Speicherpuffer befinden, ebenfalls in den Dateipuffer übertragen.

Nach Wiederherstellung der Verbindung zum Zielsystem werden immer die ältesten Daten zuerst gesendet. Neuere Werte werden währenddessen dem Puffer hinzugefügt. Wenn beim Start der Erfassung noch gepufferte Daten im Dateipuffer vorhanden sind, werden diese genauso berücksichtigt und verarbeitet. Die Daten werden in dem Format gespeichert, das zum Zeitpunkt der Pufferung in der Datenaufzeichnung konfiguriert war, und sie werden auch in diesem Format bei Wiederherstellung der Verbindung gesendet.

Im Knoten *Puffer* der jeweiligen Datenaufzeichnung nehmen Sie die Konfiguration der Pufferung vor. Die nachfolgende Abbildung zeigt die Pufferkonfiguration am Beispiel der zeitbasierten Datenaufzeichnung.



### Speicherpuffer

Der Speicherpuffer ist immer aktiviert. Sie können ihn nicht deaktivieren, da zu übertragende Daten immer den Puffer durchlaufen, bevor sie an das Zielsystem weitergegeben werden.

### Maximale Größe

Geben Sie hier die maximale Gesamtgröße für im Speicher gepufferte Elemente ein. Wenn die maximale Größe überschritten wird, gibt es 2 Möglichkeiten:

- Wenn die Dateipufferung deaktiviert ist, wird das älteste Element im Speicher gelöscht (und ist damit für immer verloren).
- Wenn die Dateipufferung aktiviert ist, wird der älteste Teil des Pufferspeichers in eine Pufferdatei verschoben.

### Speicherpuffer regelmäßig speichern alle ... s

Diese Option können Sie nur aktivieren, wenn der Dateipuffer aktiviert ist. Wenn die Option aktiviert ist, wird der gesamte Speicherpuffer periodisch in eine Pufferdatei ausgelagert.

Geben Sie eine Dauer ein, nach der der Speicherpuffer regelmäßig gesichert wird. Sie muss zwischen 10 s und 600 s liegen.

Mit dieser Option können Sie erreichen, dass möglichst wenig Daten bei einem Systemausfall verloren gehen.

### Aktuelle Speicherkonfiguration

Anzeige, welche Zeitspanne ungefähr mit den konfigurierten Einstellungen im Speicherpuffer zwischengespeichert werden kann. Angabe in d.hh:mm:ss.

## Dateipuffer

### Dateipuffer verwenden

Standardmäßig wird der Dateipuffer nicht verwendet. Hier können Sie die Dateipufferung aktivieren.

### Aktuelle Dateikonfiguration

Anzeige, welche Zeitspanne ungefähr mit den konfigurierten Einstellungen im Dateipuffer zwischengespeichert werden kann. Angabe in d.hh:mm:ss.

### Dateiverzeichnis

Im Feld *Dateiverzeichnis* können Sie einen Speicherort für die Dateien wählen. Sie können das Verzeichnis direkt in das Textfeld eingeben oder über den Browse-Button <...> auswählen. Das konfigurierte Dateiverzeichnis muss sich auf einer lokalen Festplatte des *ibaPDA*-Server-Rechners befinden.

Das gleiche Dateiverzeichnis können Sie für mehrere Datenaufzeichnungen verwenden, da die Pufferdateien einer Datenaufzeichnung einen eindeutigen Namen haben. Dateien aus verschiedenen Datenaufzeichnungen können somit anhand ihres Namens unterschieden werden.

### Maximale Puffergröße

Sie können die maximale Gesamtgröße der Pufferdateien einer Datenaufzeichnung konfigurieren. Die Pufferdateien selbst haben die Dateierweiterung *.buf*, die Index-Datei zur Verwaltung der Pufferdateien hat die Endung *.info*. Die maximale Größe ist die Gesamtgröße aller dieser Dateien. Wenn die maximale Puffergröße überschritten wird, wird die älteste Pufferdatei gelöscht.

### Weitere Puffereinstellungen

#### Maximale Zeit

Gespeicherte Daten, die älter als die maximale Zeit sind, werden nicht an das Zielsystem übertragen. Dateien, die älter als die Maximalzeit sind, können gelöscht werden. Sie können einen Wert zwischen 1 und 1000 Stunden eingeben.

### Diagnose Speicherpuffer/Diagnose Dateipuffer

#### Zuletzt entferntes Element

Zeigt an, wann das letzte Element aus diesem Teil des Puffers genommen wurde.

#### Füllstand

Der Füllstand gibt jeweils an, welcher prozentuale Anteil der Puffergröße aktuell mit gepufferten Daten gefüllt ist.

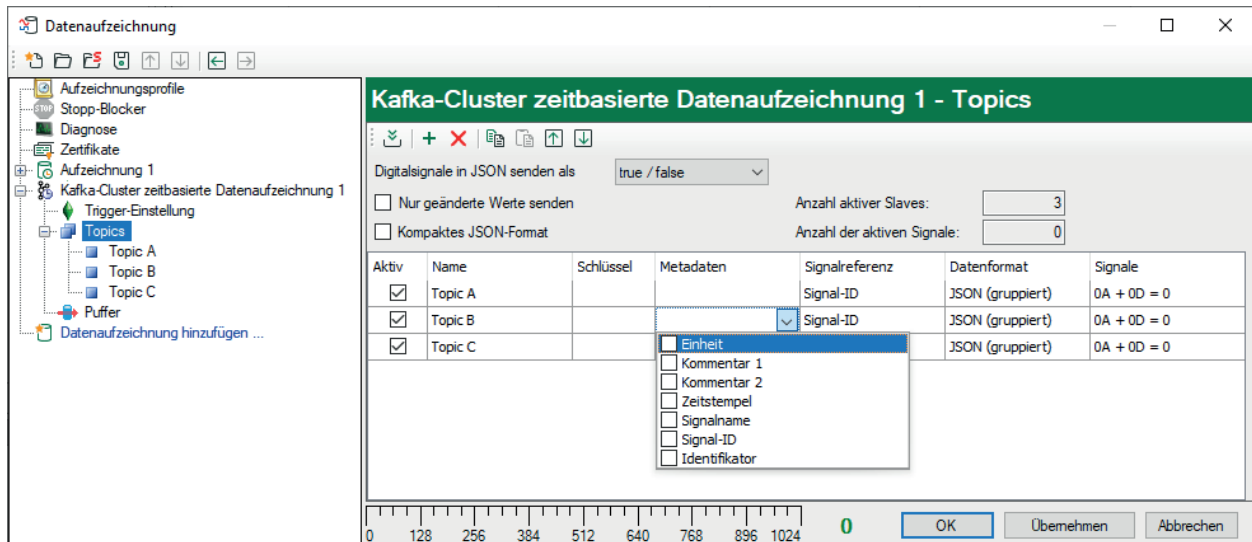
#### Unverarbeiteter Anteil

Im Dateipuffer werden Elemente, die an das Zielsystem übertragen wurden, nicht sofort gelöscht. Erst wenn eine Pufferdatei vollständig gelesen ist, wird sie gelöscht. Es ist also möglich, dass nur ein Teil einer Pufferdatei noch nicht übertragene Daten enthält. Der Füllstand bezieht sich auf die vorhandenen Pufferdateien, der "unverarbeitete Anteil" gibt dagegen den prozentualen Anteil der noch nicht übertragenen Daten im Dateipuffer wieder.

### 3.4 Konfiguration der Topics

Im Kafka-Cluster sind in der Regel mehrere Topics definiert, die vom Cluster-Administrator erstellt werden. Es ist nicht möglich Topics von *ibaPDA* aus im Cluster anzulegen oder zu löschen.

Wenn Sie den Knoten *Topics* markieren, wird die Liste der Topics angezeigt, die von der Datenaufzeichnung beschrieben werden sollen.



Buttons zur Konfiguration der Topics:

	Topic-Liste von Kafka-Cluster herunterladen (Verbindung zum Cluster erforderlich)
	Neues Topic manuell hinzufügen
	Markiertes Topic entfernen
	Markierte Zelleninhalte in die Zwischenablage kopieren
	Daten aus der Zwischenablage in Zellen einfügen
	Markiertes Topic nach oben verschieben
	Markiertes Topic nach unten verschieben

#### Hinweis



Mit den Tasten <Umsch> und <Strg> können Sie mehrere Zellen markieren und kopieren/einfügen/löschen.

Wenn Sie JSON als Datenformat wählen, können Sie einstellen, ob digitale Signale als 1/0 oder true/false gesendet werden.

Sie können die zu übertragenden Daten verringern, wenn Sie die Option *Nur geänderte Werte senden* aktivieren.

Die Option *Kompaktes JSON-Format* verringert die übertragene Datenmenge ebenfalls und entfernt Füllzeichen, wie z. B. mehrfache Leerzeichen, Zeilenumbrüche usw., aus dem Telegramm.

In den Feldern rechts daneben wird die Anzahl der aktiven Topics und der aktiven Signale angezeigt.

## Bedeutung der Spalten

### Aktiv

Hier aktivieren/deaktivieren Sie das Schreiben in ein Topic.

### Name

Geben Sie hier einen eindeutigen Namen für das Topic ein.

### Schlüssel

Die Eigenschaft *Schlüssel* ist optional. Sie können diese Eigenschaft durch eine Reihe von Platzhaltern definieren:

- \$identifier: Identifikator, definiert in der Datenaufzeichnungskonfiguration, siehe Kapitel [↗ Datenaufzeichnung Kafka-Cluster](#), Seite 10
- \$signalid: Signal-ID
- \$signalname: Signalname
- \$unit: Signaleinheit
- \$comment1: Signalkommentar 1
- \$comment2: Signalkommentar 2

Bei Verwendung des gruppierten Datenformats *JSON (gruppiert)* werden signalbezogene Platzhalter durch leeren Text ersetzt.

### Metadaten

Wenn Sie für das Topic eines der JSON-Formate als Datenformat gewählt haben, können Sie im Drop-down-Menü Metadaten auswählen, die mit dem Topic gespeichert werden.

Zur Auswahl stehen: Einheit, Kommentar 1 + 2, Zeitstempel, Signalname, Signal-ID, Identifikator. Markieren Sie die gewünschten Metadaten mit einem Häkchen.

---

### Hinweis



Die Metadaten sind in jeder Nachricht enthalten. Da alle Metadaten außer dem Zeitstempel während einer Messung konstant sind, werden viele redundante Daten in den Cluster geschrieben. Es ist daher wichtig abzuwägen, welche Metadaten tatsächlich erforderlich sind.

Der Zeitstempel ist implizit bereits in jeder Nachricht enthalten. Das Hinzufügen des *Zeitstempel* zu den Metadaten einer Nachricht führt daher auch zu redundanter Information. Aus Sicht der Datenanalyse kann es trotzdem sinnvoll sein, den *Zeitstempel* in den Metadaten zu aktivieren.

---

## Signalreferenz

Wählen Sie im Drop-down-Menü, ob die Signal-ID oder der Signalname als Signalreferenz verwendet werden soll.

## Datenformat

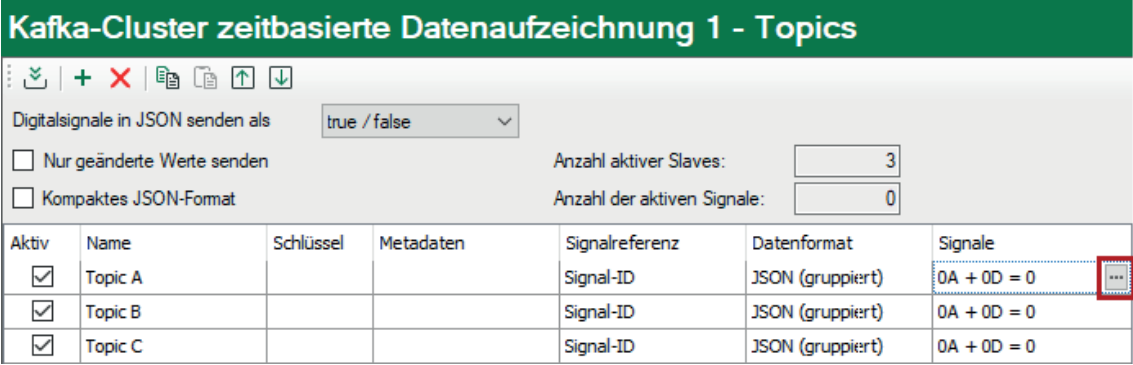
Folgende Datenformate werden unterstützt:

- *JSON (gruppiert)*: alle Signaldaten für einen bestimmten Zeitstempel werden in einer Nachricht zusammengefasst
- *JSON (pro Signal)*: eine Nachricht wird pro Signal und Zeitstempel erstellt
- *AVRO (pro Signal)*: eine Nachricht wird pro Signal und Zeitstempel erstellt

Beispiele für die Beschreibung der Datenformate finden Sie im Kapitel [↗ Beschreibung der Datenformate](#), Seite 31.

## Signale

Um die Signale zu konfigurieren, die Sie in ein Topic schreiben wollen, markieren Sie das Topic im Strukturbaum oder klicken auf den Button <...> in der Spalte *Signale*.



Aktiv	Name	Schlüssel	Metadaten	Signalreferenz	Datenformat	Signale
<input checked="" type="checkbox"/>	Topic A			Signal-ID	JSON (gruppiert)	0A + 0D = 0
<input checked="" type="checkbox"/>	Topic B			Signal-ID	JSON (gruppiert)	0A + 0D = 0
<input checked="" type="checkbox"/>	Topic C			Signal-ID	JSON (gruppiert)	0A + 0D = 0

Im folgenden Dialog ordnen Sie die gewünschten Signale unter Nutzung der Aufzeichnungsprofile den *Topics* zu. Siehe Kapitel [↗ Signalauswahl](#), Seite 23.

## 4 Signalauswahl

Um Signale aufzuzeichnen, müssen Sie die Signale mithilfe eines Aufzeichnungsprofil vom Typ *Zeit* einem Topic zuordnen.

Klicken Sie im Knoten *Topics* in der Spalte *Signale* der Topic-Liste auf den Button <...>, um zum Signalauswahldialog zu gelangen.

**Kafka-Cluster zeitbasierte Datenaufzeichnung 1 - Topics**

Digitalsignale in JSON senden als:

☐ Nur geänderte Werte senden

☐ Kompaktes JSON-Format

Anzahl aktiver Slaves:

Anzahl der aktiven Signale:

Aktiv	Name	Schlüssel	Metadaten	Signalreferenz	Datenformat	Signale
<input checked="" type="checkbox"/>	Topic A			Signal-ID	JSON (gruppiert)	0A + 0D = 0 <span style="border: 1px solid red; padding: 2px;">...</span>
<input checked="" type="checkbox"/>	Topic B			Signal-ID	JSON (gruppiert)	0A + 0D = 0
<input checked="" type="checkbox"/>	Topic C			Signal-ID	JSON (gruppiert)	0A + 0D = 0

Oder markieren Sie ein Topic in der Baumstruktur.

**Datenaufzeichnung**

**Topics - Topic A**

**Profile**

Name	Signale
As is	0A + 0D = 0
<b>Zeit Kafka</b>	<b>36A + 1D = 37</b>

**Signale**

- ☒ 0. Hydr. Anstellung
- ☐ 1. Schere / RSF / S1-S6
- ☐ 2. Gerüste 1-7 u Walzkraft
- ☐ 2:0: F5 Servo BS 1
- ☐ 2:1: F5 Servo BS 2
- ☐ 2:2: F5 Servo AS 1
- ☐ 2:3: F5 Servo AS 2
- ☒ 2:4: 060 F1 Drehzahl
- ☒ 2:5: 061 F1 Strom
- ☒ 2:6: 062 F2 Drehzahl
- ☒ 2:7: 063 F2 Strom
- ☐ 2:8: 064 F3 Drehzahl
- ☐ 2:9: 065 F3 Strom
- ☐ 2:10: 066 F4 Drehzahl
- ☐ 2:11: 067 F4 Strom
- ☐ 2:12: 068 F5 Drehzahl
- ☐ 2:13: 069 F5 Strom
- ☐ 2:14: 070 F6 Drehzahl
- ☐ 2:15: 071 F6 Strom
- ☐ 2:16: 072 F7 Drehzahl
- ☐ 2:17: 073 F7 Strom
- ☐ 2:18: 074 F1 Walzkraft AS
- ☐ 2:19: 075 F1 Walzkraft BS
- ☐ 2:20: 076 F2 Walzkraft AS
- ☐ 2:21: 077 F2 Walzkraft BS

Profileigenschaften  
Modus: Original-Zeitbasis  
Komprimierung: Standard

0 128 256 384 512 640 768 896 1024 **37**

### Hinweis



Nähere Informationen zu den Aufzeichnungsprofilen finden Sie im Handbuch *ibaPDA*, Teil 5.

Markieren Sie das Topic, dem Sie bestimmte Signale zuordnen möchten, und wählen Sie in der Profilliste ein Aufzeichnungsprofil. Setzen Sie ein Häkchen in die Auswahlfelder neben den Signalen, denen Sie dieses Profil zuordnen möchten.

Im Abschnitt *Profileigenschaften* werden einige Informationen zur konfigurierten Zeitbasis und Filterung des ausgewählten Profils angezeigt.

Kafka-Cluster-Datenaufzeichnungen werden pro Anzahl der geschriebenen Signale lizenziert. Die aktuelle Anzahl der ausgewählten Signale in allen Kafka-Cluster-Datenaufzeichnungen finden Sie am unteren Rand des Dialogs angezeigt, ähnlich wie die Anzahl der konfigurierten Signale im I/O-Manager. Die Länge des Balkens entspricht der lizenzierten Signalanzahl.

Im obigen Beispiel können maximal 1024 Signale über Kafka-Datenaufzeichnungen geschrieben werden. Derzeit sind 37 Signale aktiviert.

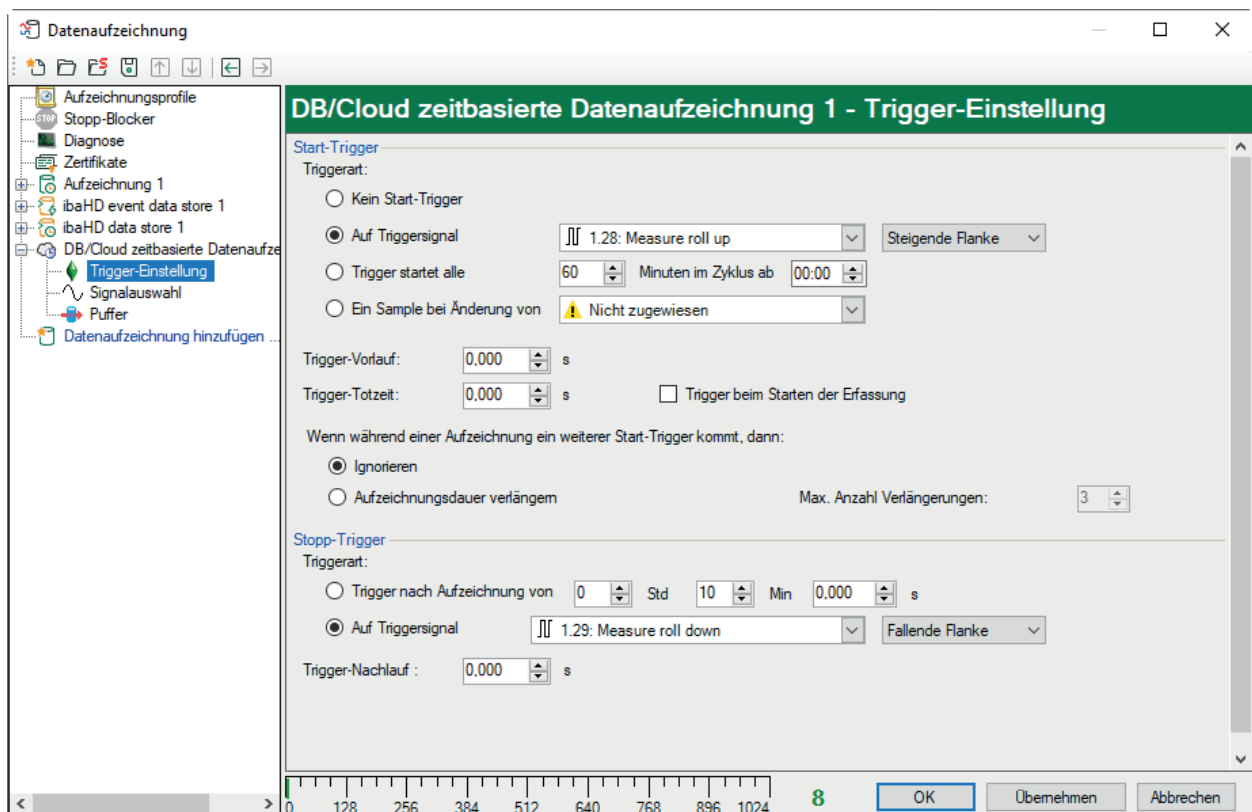


## 5 Trigger-Einstellung

Die Beschreibung gilt für folgende Typen von Datenaufzeichnungen:

- DB/Cloud zeitbasiert
- Kafka-Cluster zeitbasiert
- MQTT zeitbasiert und MQTT Sparkplug B
- MindSphere zeitbasiert
- InfluxDB zeitbasiert

Im Knoten *Trigger-Einstellungen* legen Sie fest, wann Daten aufgezeichnet werden, hier am Beispiel *DB/Cloud zeitbasierte Datenaufzeichnung*.



### Start-Trigger

Zunächst können Sie wählen, ob Sie kontinuierlich aufzeichnen möchten oder ausgelöst durch einen Trigger.

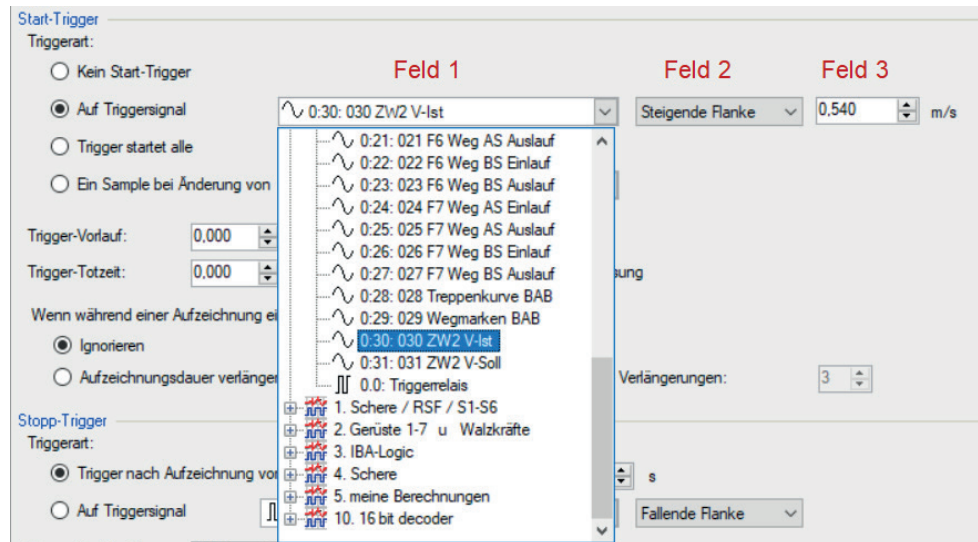
### Kein Start-Trigger

Mit dieser Option werden die Daten kontinuierlich aufgezeichnet. Der Beginn der Messaufzeichnung erfolgt sofort bei Start der Messung, bzw. über den Button <GO>.

## Auf Triggersignal

Wenn Sie ein Messsignal oder ein virtuelles Signal als Trigger verwenden wollen, wählen Sie die Option *Auf Triggersignal*. In den Feldern daneben legen Sie die Eigenschaften des Triggersignals fest.

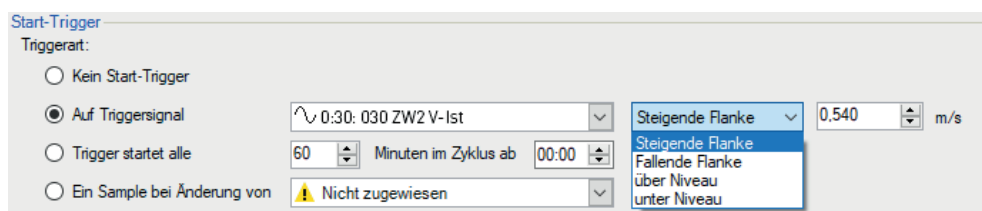
- Feld 1: Auswahlliste zur Signalauswahl (verfügbare Analogsignale und Digitalsignale)
- Feld 2: Auswahlliste zur Auswahl von Flanken und Pegeln
- Feld 3: Auswahlliste zur Auswahl des Triggerpegels, in der jeweiligen physikalischen Einheit (Feld 3 ist nur bei analogen Triggersignalen aktiv)



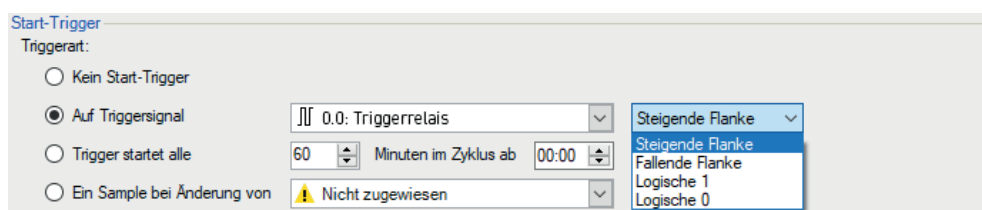
Sie können Analogsignale und Digitalsignale als Trigger verwenden. Wählen Sie über den Signalbaum in der Auswahlliste in Feld 1 das Signal, auf das getriggert werden soll.

In den Felder 2 und 3 können Sie das Triggerereignis genauer definieren. Diese Felder variieren je nachdem, ob der ausgewählte Messkanal analog oder digital ist.

- Bei Analogsignalen können Sie wählen zwischen Niveau-Triggerern und Flanken-Triggerern einschließlich eines vorgegebenen Pegels (Feld 3).



- Bei Digitalsignalen können Sie wählen zwischen Pegel-Triggerern und Flanken-Triggerern unter Einbeziehung der 2 Pegel logisch 0 (FALSE) und logisch 1 (TRUE).



**Trigger startet alle ...**

Wenn Sie regelmäßig einen Start-Trigger immer zu einer bestimmten Tageszeit verwenden möchten, können Sie die Option *Trigger startet alle ... Minuten im Zyklus ab ...* wählen. Geben Sie den Zeitraum in Minuten ein oder wählen Sie ihn über das Eingabefeld. Der Wertebereich liegt zwischen 0 und 1440, was einem Tag entspricht. Danach wählen Sie eine Start-Zeit für den ersten Trigger. Der Wertebereich liegt zwischen 00:00 und 23:59, was einem Tag entspricht.

**Ein Sample bei Änderung von**

Wenn sich der Wert des ausgewählten Signals ändert, wird ein Messwert aufgezeichnet. Die Aufzeichnung stoppt nach einem Messwert, bis die nächste Signaländerung erkannt wird. Eine Totzeit kann konfiguriert werden, um eine minimale Zeitspanne zwischen den Messwerten festzulegen. Bevor die Totzeit nicht abgelaufen ist, wird kein neuer Messwert aufgezeichnet.

**Trigger-Vorlauf**

Sie können eine Trigger-Vorlaufzeit konfigurieren, dann beginnt die Aufzeichnung, um die Trigger-Vorlaufzeit vor dem Triggerereignis. Wird die Triggerbedingung erfüllt, werden die einlaufenden Daten an die während der Vorlaufzeit gepufferten Daten angehängt.

**Trigger-Totzeit**

Diese Eigenschaft steht bei den Start-Trigger *Auf Triggersignal*, *Trigger startet alle ...* und *Ein Sample bei Änderung von* zur Verfügung. Die Trigger-Totzeit gibt an, wie lange nach dem Auftreten eines Triggers alle nachfolgenden Trigger unterdrückt werden.

Ist die Totzeit beispielsweise eingestellt auf 5 Sekunden, dann werden in den 5 Sekunden nach dem Auftreten des ersten Triggers alle anderen Trigger ignoriert.

**Trigger beim Starten der Erfassung**

Wenn die Aufzeichnung direkt beim Start der Erfassung starten soll oder sobald eine neue Datenaufzeichnungs-Konfiguration übernommen wurde, dann muss zusätzlich die Option *Trigger beim Starten der Erfassung* aktiviert werden. Wenn Sie die Option nicht aktivieren, startet die Aufzeichnung erst, wenn der Trigger auslöst.

**Wenn während einer Aufzeichnung ein weiterer Start-Trigger kommt, dann:**

Hier können Sie festlegen, was passieren soll, wenn ein neuer Starttrigger auftritt, während eine Aufzeichnung bereits läuft.

- Ignorieren:  
Ist diese Option aktiviert, dann ignoriert das System während einer laufenden Aufzeichnung jeden neuen Start-Trigger solange, bis der Stopp-Trigger für diese Aufzeichnung auftritt.
- Aufzeichnungsdauer verlängern:  
Ist diese Option aktiviert, dann verlängert das System bei Auftreten eines weiteren Start-Triggers während einer laufenden Aufzeichnung die Dauer der laufenden Aufzeichnung. Dies geschieht so oft, wie im Feld "Maximale Anzahl Verlängerungen" eingestellt wurde. Ist die max. Anzahl Verlängerungen erreicht, werden alle weiteren Start-Trigger ignoriert. Natürlich beendet jeder vorher auftretende Stopp-Trigger die Aufzeichnung sofort.

**Stopp-Trigger**

Die Einstellungen für den Stopp-Trigger erfolgen auf die gleiche Weise wie des Start-Triggers. Auch hier können Sie sowohl analoge als auch digitale Signale als Trigger verwendet.

**Trigger nach Aufzeichnung von x Std x Min x s**

Hier können Sie eine Zeitspanne konfigurieren, nach welcher die Aufzeichnung beendet wird – ab Auftreten des Start-Triggers.

**Auf Triggersignal**

Siehe Erläuterungen unter Start-Trigger.

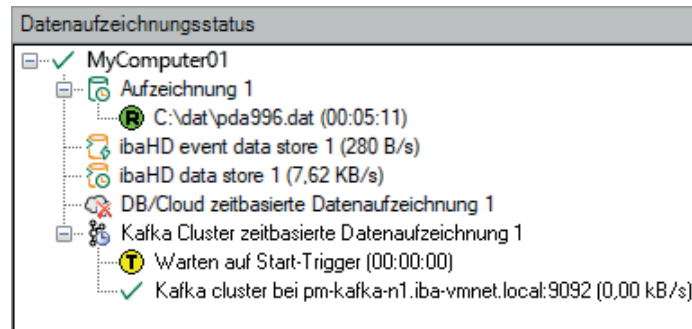
**Trigger-Nachlauf**

Sie können eine Trigger-Nachlaufzeit konfigurieren, dann endet die Aufzeichnung um die Trigger-Nachlaufzeit nach dem Stopp-Triggerereignis.

## 6 Diagnose

### 6.1 Status der Datenaufzeichnung

Der aktuelle Status der Datenaufzeichnungen ist im Statusfenster der Datenaufzeichnung ersichtlich.



Hier werden alle definierten Datenaufzeichnungen und ihr jeweiliger Status angezeigt, je nach Datenaufzeichnung mit Server-Adresse, Aufzeichnungsdauer, Schreibgeschwindigkeit, etc.

Das Symbol vor dem Namen zeigt den aktuellen Zustand der Aufzeichnung an:

- ..... Warten auf den Start-Trigger (nur bei getriggelter Aufzeichnung)
  - ..... Aufzeichnung läuft
  - ..... Triggernachlauf, Stopp-Trigger ist erfolgt, aber Aufzeichnung läuft noch weiter für die Dauer des Triggernachlaufs
- Deaktivierte oder fehlerhafte Datenaufzeichnungen werden mit einem roten Kreuz im Symbol der Datenaufzeichnung gekennzeichnet.

Mit einem Rechtsklick auf diesen Knoten können Sie auch manuell einen Start- oder Stopptrigger senden.

## 6.2 Diagnose der Datenaufzeichnung

Der Knoten *Diagnose* im Baum der Datenaufzeichnungen bietet Informationen zur Belastung des Systems durch die Datenaufzeichnungen. Die Messung muss dazu laufen.

**Diagnose**

Gesamtlast im Erfassungsthread durch Datenaufzeichnungen: 0,38% Statistik zurücksetzen

Datenaufzeichnung		Schreibgeschwindigke...		Speicherpuffer (kB)		Dateipuffer (MB)		Erfassung Thread-Last
Name	Ziel	Mittelwert	Max	Mittelwert	Max	Mittelwert	Max	
Erfassungsthread (0,38%)								
Aufzeichnung 1	C:\	15,15	332,90	0,00	0,00			0,18%
ibaHD event data store 1 (0,88%)								
ibaHD event data sto...	MyComputer01\HD_TRIG	0,27	0,39	0,00	0,01			0,02%
ibaHD data store 1 (0,20%)								
ibaHD data store 1	MyComputer01\HD_Time2	7,02	9,16	0,00	1,56			0,04%
Kafka-Cluster zeitbasierte Datenaufzeichnung 1 (0,00%)								
Kafka-Cluster zeitbas...	localhost:9092	0,00	0,00	3850,62	3878,03			0,02%
MQTT zeitbasierte Datenaufzeichnung 1 (0,00%)								
MQTT zeitbasierte Da...	localhost	0,00	0,00	267,01	288,26			0,05%

Einheit der Schreibgeschwindigkeit: ☒ kB/s ☐ MB/s ☐ MB/h

OK Übernehmen Abbrechen

In der Tabelle werden die Performanzwerte aller Datenaufzeichnungen dargestellt. Pro Datenaufzeichnung gibt es eine Zeile. Die Zeilen sind gruppiert nach den Threads, die die Daten schreiben.

In jeder Gruppenzeile steht der Name des Threads und (in Klammern) die Auslastung (100% bedeutet, dass der Thread überlastet ist). Standardmäßig wird der Mittelwert der Last angezeigt. Sie können aber über das Kontextmenü zwischen Mittelwert und Istwert umschalten.

Die Spalte *Ziel* zeigt das jeweilige Ziel an, in das die Daten geschrieben werden, beispielsweise eine Festplattenpartition, die Adresse der Datenbank, die Adresse des Kafka-Clusters etc. Die *Schreibgeschwindigkeit* zeigt an, wie schnell die Daten geschrieben werden.

Die Spalten *Speicherpuffer* zeigen an, wie viele Daten in *ibaPDA* gepuffert werden. Die Spalten *Dateipuffer (MB)* zeigen an, wie viele Daten im Dateipuffer gepuffert werden.

Die Spalte *Erfassung Thread-Last* zeigt je nach Datenaufzeichnung unterschiedliche Informationen. Bei zeitbasierten Datenaufzeichnungen zeigt die Spalte *Erfassung Thread-Last* den zeitlichen Anteil, der für die Lauflängenkodierung und das Schreiben auf Platte benötigt wird.

Bei DB/Cloud-, MQTT-, Kafka-Cluster-, InfluxDB- und MindSphere-Datenaufzeichnungen zeigt die Spalte die Last, die durch die Auswertung der Trigger und die Erzeugung der Zeilendaten verursacht wird.

Bei ibaHD-Datenaufzeichnungen wird jeweils die anteilige Verarbeitungszeit angezeigt, die für die Erzeugung der auf den HD-Server zu schreibenden Daten aufgewendet wird. In diesen Werten sind bereits die Lauflängenkodierung für zeitbasierte Aufzeichnungen, Ereignis-Trigger-Berechnung für ereignisbasierte Aufzeichnungen und die Berechnung der längenbezogenen Daten für längenbasierte Aufzeichnungen enthalten.

Weitere Informationen zur Diagnose finden Sie im Handbuch *ibaPDA*, Teil 5.

## 7 Anhang

### 7.1 Beschreibung der Datenformate

#### 7.1.1 Datenformat JSON (gruppiert)

Im Folgenden wird der Aufbau des Datenformats "JSON (gruppiert)" anhand eines Beispiels mit drei Signalen erläutert. Die Signale sind wie folgt definiert:

Virtual (0)					
<div> <span> Allgemein</span> <span> Analog</span> <span> Digital</span> </div>					
	Name	Ausdruck	Einheit	Kommentar 1	Kommentar 2
0	Signalname_0	GenerateSignal(0)	Einheit_0	Beispielkommentar1_0	Beispielkommentar2_0
1	Signalname_1	GenerateSignal(1)	Einheit_0	Beispielkommentar1_1	Beispielkommentar2_1
2	Signalname_2	GenerateSignal(2)	Einheit_0	Beispielkommentar1_2	Beispielkommentar2_2
*					

#### Beispiel JSON bei Signal-ID als Signalreferenz

```
{
  "[0:0]": Istwert,
  "[0:0].ID": "[0:0]",
  "[0:0].Name": "Signalname_0",
  "[0:0].Unit": "Einheit_0",
  "[0:0].Comment1": "Beispielkommentar1_0",
  "[0:0].Comment2": "Beispielkommentar2_0",
  "[0:1]": Istwert,
  "[0:1].ID": "[0:1]",
  "[0:1].Name": "Signalname_1",
  "[0:1].Unit": "Einheit_1",
  "[0:1].Comment1": "Beispielkommentar1_1",
  "[0:1].Comment2": "Beispielkommentar2_1",
  "[0:2]": Istwert,
  "[0:2].ID": "[0:2]",
  "[0:2].Name": "Signalname_2",
  "[0:2].Unit": "Einheit_2",
  "[0:2].Comment1": "Beispielkommentar1_2",
  "[0:2].Comment2": "Beispielkommentar2_2",
  "Timestamp": "2020-01-21T13:10:53.0002189Z",
  "Identifizier": "Mein Identifikator"
}
```

Rot: optionale signalbezogene Metadaten

Grün: optionale gruppenbezogene Metadaten

**Beispiel JSON bei Signalname als Signalreferenz**

```
{
  "Signalname_0": Actual value,
  "Signalname_0.ID": "[0:0]",
  "Signalname_0.Name": "Signalname_0",
  "Signalname_0.Unit": "Unit_0",
  "Signalname_0.Comment1": "Example_comment1_0",
  "Signalname_0.Comment2": "Example_comment2_0",
  "Signalname_1": Actual value,
  "Signalname_1.ID": "[0:1]",
  "Signalname_1.Name": "Signalname_1",
  "Signalname_1.Unit": "Unit_1",
  "Signalname_1.Comment1": "Example_comment1_1",
  "Signalname_1.Comment2": "Example_comment2_1",
  "Signalname_2": Actual value,
  "Signalname_2.ID": "[0:2]",
  "Signalname_2.Name": "Signalname_2",
  "Signalname_2.Unit": "Unit_2",
  "Signalname_2.Comment1": "Example_comment1_2",
  "Signalname_2.Comment2": "Example_comment2_2",
  "Timestamp": "2020-01-21T13:20:13.0009119Z",
  "Identifier": "My identifier"
}
```

Rot: optionale signalbezogene Metadaten

Grün: optionale gruppenbezogene Metadaten



### 7.1.2 Datenformat JSON (pro Signal)

Im Folgenden wird der Aufbau des Datenformats "JSON (pro Signal)" anhand eines Beispiels mit drei Signalen erläutert. Die Signale sind wie folgt definiert:

Virtual (0)					
<div> <span> Allgemein</span> <span> Analog</span> <span> Digital</span> </div>					
	Name	Ausdruck	Einheit	Kommentar 1	Kommentar 2
0	Signalname_0	GenerateSignal(0)	Einheit_0	Beispielkommentar1_0	Beispielkommentar2_0
1	Signalname_1	GenerateSignal(1)	Einheit_0	Beispielkommentar1_1	Beispielkommentar2_1
2	Signalname_2	GenerateSignal(2)	Einheit_0	Beispielkommentar1_2	Beispielkommentar2_2
*					

#### Beispiel JSON bei Signal-ID als Signalreferenz

```
{
  "Signal": "[0:0]",
  "Value": Actual value,
  "ID": "[0:0]",
  "Name": "Signalname_0",
  "Unit": "Unit_0",
  "Comment1": "Example_comment1_0",
  "Comment2": "Example_comment2_0",
  "Timestamp": "2020-01-21T13:26:50.8784074Z",
  "Identifier": "My identifier"
}

{
  "Signal": "[0:1]",
  "Value": Actual value,
  "ID": "[0:1]",
  "Name": "Signalname_1",
  "Unit": "Unit_1",
  "Comment1": "Example_comment1_1",
  "Comment2": "Example_comment2_1",
  "Timestamp": "2020-01-21T13:26:50.8784074Z",
  "Identifier": "My identifier"
}

{
  "Signal": "[0:2]",
  "Value": Actual value,
  "ID": "[0:2]",
  "Name": "Signalname_2",
  "Unit": "Unit_2",
  "Comment1": "Example_comment1_2",
  "Comment2": "Example_comment2_2",
  "Timestamp": "2020-01-21T13:26:50.8784074Z",
  "Identifier": "My identifier"
}
```

Rot: optionale signalbezogene Metadaten

**Beispiel JSON bei Signalname als Signalreferenz**

```
{
  "Signal": "Signalname_0",
  "Value": Actual value,
  "ID": "[0:0]",
  "Name": "Signalname_0",
  "Unit": "Unit_0",
  "Comment1": "Example_comment1_0",
  "Comment2": "Example_comment2_0",
  "Timestamp": "2020-01-21T13:36:37.5310016Z",
  "Identifier": "My identifier"
}

{
  "Signal": "Signalname_1",
  "Value": Actual value,
  "ID": "[0:1]",
  "Name": "Signalname_1",
  "Unit": "Unit_1",
  "Comment1": "Example_comment1_1",
  "Comment2": "Example_comment2_1",
  "Timestamp": "2020-01-21T13:36:37.5310016Z",
  "Identifier": "My identifier"
}

{
  "Signal": "Signalname_2",
  "Value": Actual value,
  "ID": "[0:2]",
  "Name": "Signalname_2",
  "Unit": "Unit_2",
  "Comment1": "Example_comment1_2",
  "Comment2": "Example_comment2_2",
  "Timestamp": "2020-01-21T13:36:37.5310016Z",
  "Identifier": "My identifier"
}
```

Rot: optionale signalbezogene Metadaten

### 7.1.3 Datenformat AVRO (pro Signal)

Im Vergleich zu JSON, das Daten in einem lesbaren Format speichert, verwendet AVRO eine binäre Kodierung, die die Bandbreite und den benötigten Speicherplatz reduziert.

*ibaPDA* verwendet das folgende Schema zur Serialisierung der Signaldaten:

```
{
  "namespace": "de.iba",
  "type": "record",
  "name": "PdaRecord",
  "fields": [
    {"name": "Signal", "type": "string"},
    {"name": "ID", "type": ["null", "string"]},
    {"name": "Name", "type": ["null", "string"]},
    {"name": "Unit", "type": ["null", "string"]},
    {"name": "Comment1", "type": ["null", "string"]},
    {"name": "Comment2", "type": ["null", "string"]},
    {"name": "Timestamp", "type": [
      "null",
      {"type": "long", "logicalType": "timestamp-micros"}
    ]},
    {"name": "Identifier", "type": ["null", "string"]},
    {"name": "ValueType", "type": {
      "type": "enum",
      "name": "ValueTypeEnum",
      "symbols": ["BOOLEAN", "BYTES", "DOUBLE", "FLOAT", "INT", "LONG", "STRING"]
    }},
    {"name": "BooleanValue", "type": ["null", "boolean"]},
    {"name": "BytesValue", "type": ["null", "bytes"]},
    {"name": "DoubleValue", "type": ["null", "double"]},
    {"name": "FloatValue", "type": ["null", "float"]},
    {"name": "IntValue", "type": ["null", "int"]},
    {"name": "LongValue", "type": ["null", "long"]},
    {"name": "StringValue", "type": ["null", "string"]}
  ]
}
```

*ibaPDA* unterstützt auch eine Verbindung zu einer Confluent Schema Registry. Dies ist jedoch nicht unbedingt erforderlich, sodass dieses Feld leer gelassen werden kann. Wenn eine Schema Registry konfiguriert wird, registriert *ibaPDA* das für die Kodierung verwendete Schema in der Schema Registry. Die zurückgegebene ID wird dann wie folgt an jeden Datensatz angehängt:

Byte Offset	Beschreibung
0	0x00 (Confluent AVRO Marker)
1	Schema ID (Big Endian)
5	Signaldaten

Die Adresse und der Port der Schema Registry können im Feld Schema Registry-Adresse eingegeben werden.

Mit dem Button <Verbindung testen> können Sie die Verbindung zur Schema Registry testen.

Wenn keine Schema Registry verwendet wird, wird jeder Datensatz wie folgt kodiert:

Byte Offset	Beschreibung
0	0xC3 0x01 (Single-object encoding marker)
2	CRC-64-AVRO fingerprint of encoding schema
10	Signaldaten

#### Hinweis



Detaillierte Informationen zu AVRO finden Sie hier:

<https://avro.apache.org/docs/>

## 8 Support und Kontakt

### Support

Tel.: +49 911 97282-14  
E-Mail: support@iba-ag.com

---

### Hinweis



Wenn Sie Support benötigen, dann geben Sie bitte bei Softwareprodukten die Nummer des Lizenzcontainers an. Bei Hardwareprodukten halten Sie bitte ggf. die Seriennummer des Geräts bereit.

---

### Kontakt

#### Hausanschrift

iba AG  
Königswarterstraße 44  
90762 Fürth  
Deutschland

Tel.: +49 911 97282-0  
E-Mail: iba@iba-ag.com

#### Postanschrift

iba AG  
Postfach 1828  
90708 Fürth

#### Warenanlieferung, Retouren

iba AG  
Gebhardtstraße 10  
90762 Fürth

#### Regional und weltweit

Weitere Kontaktadressen unserer regionalen Niederlassungen oder Vertretungen finden Sie auf unserer Webseite:

**[www.iba-ag.com](http://www.iba-ag.com)**