ibaPDA-Data-Store-Kafka
Data streaming into Apache Kafka cluster

Manual
Issue 1.1

Measurement Systems for Industry and Energy
www.iba-ag.com
Manufacturer
iba AG
Koenigswarterstr. 44
90762 Fuerth
Germany

Contacts
Main office +49 911 97282-0
Fax +49 911 97282-33
Support +49 911 97282-14
Engineering +49 911 97282-13
E-mail iba@iba-ag.com
Web www.iba-ag.com

Unless explicitly stated to the contrary, it is not permitted to pass on or copy this document, nor to make use of its contents or disclose its contents. Infringements are liable for compensation.
© iba AG 2020, All rights reserved.
The content of this publication has been checked for compliance with the described hardware and software. Nevertheless, discrepancies cannot be ruled out, and we do not provide guarantee for complete conformity. However, the information furnished in this publication is updated regularly. Required corrections are contained in the following regulations or can be downloaded on the Internet.
The current version is available for download on our web site www.iba-ag.com.

Version | Date   | Revision - Chapter / Page | Author | Version SW
--------|--------|---------------------------|--------|-----------
1.1     | 04-2020| Cluster address           |        | st 7.1.0  

Windows® is a brand and registered trademark of Microsoft Corporation. Other product and company names mentioned in this manual can be labels or registered trademarks of the corresponding owners.
# Content

1. **About this manual** .............................................................................................................4
   1.1 Target group and previous knowledge ..................................................................4
   1.2 Notations ...............................................................................................................5
   1.3 Used symbols .........................................................................................................6

2. **Introduction** ......................................................................................................................7
   2.1 System requirements .............................................................................................8

3. **Data store configuration** ..................................................................................................9
   3.1 Add a data store ....................................................................................................9
   3.2 Kafka cluster data store ......................................................................................10
   3.3 Configuring topics ..............................................................................................13

4. **Signal selection** ............................................................................................................16

5. **Trigger mode** ...................................................................................................................18

6. **Diagnostics** ......................................................................................................................22
   6.1 Data storage status ...............................................................................................22
   6.2 Diagnostics of data stores ...................................................................................23

7. **Appendix** ........................................................................................................................24
   7.1 Description of the data formats............................................................................24
      7.1.1 Data format JSON (grouped) ........................................................................24
      7.1.2 Data format JSON (per signal) ......................................................................26
      7.1.3 Data format AVRO (per signal) ....................................................................29

8. **Support and contact** ........................................................................................................31
1 About this manual

This documentation describes the function and application of the data store *ibaPDA-Data-Store-Kafka*.

This documentation is a supplement to the *ibaPDA* manual. Information about all the other characteristics and functions of *ibaPDA* may be found in the *ibaPDA* manual or in the online help.

You will find basic information about data storage in *ibaPDA* in the *ibaPDA* manual part 5.

1.1 Target group and previous knowledge

This documentation addresses qualified professionals, who are familiar with handling electrical and electronic modules as well as communication and measurement technology. A person is regarded as professional if he/she is capable of assessing the work assigned to him/her and recognizing possible risks on the basis of his/her specialist training, knowledge and experience and knowledge of the standard regulations.

This documentation in particular addresses persons, who are concerned with the configuration, test, commissioning or maintenance of the supported database, cloud or cluster storage technology. For the handling of *ibaPDA-Data-Store-Kafka* the following basic knowledge is required and/or useful:

- Windows operating system
- Basic knowledge of *ibaPDA*
- Basic knowledge of databases, cloud or cluster storage technology
### 1.2 Notations

In this manual, the following notations are used:

<table>
<thead>
<tr>
<th>Action</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu command</td>
<td>Menu <em>Logic diagram</em></td>
</tr>
<tr>
<td>Calling the menu command</td>
<td><em>Step 1 – Step 2 – Step 3 – Step x</em></td>
</tr>
<tr>
<td></td>
<td>Example: Select the menu <em>Logic diagram</em> Add - <em>New function block</em></td>
</tr>
<tr>
<td>Keys</td>
<td><em>&lt;Key name&gt;</em></td>
</tr>
<tr>
<td></td>
<td>Example: <em>&lt;Alt&gt;</em>; <em>&lt;F1&gt;</em></td>
</tr>
<tr>
<td>Press the keys simultaneously</td>
<td><em>&lt;Key name&gt;</em> + <em>&lt;Key name&gt;</em></td>
</tr>
<tr>
<td></td>
<td>Example: <em>&lt;Alt&gt;</em> + <em>&lt;Ctrl&gt;</em></td>
</tr>
<tr>
<td>Buttons</td>
<td><em>&lt;Key name&gt;</em></td>
</tr>
<tr>
<td></td>
<td>Example: <em>&lt;OK&gt;</em>; <em>&lt;Cancel&gt;</em></td>
</tr>
<tr>
<td>File names, paths</td>
<td>&quot;<em>Filename</em>, &quot;<em>Path&quot;</em></td>
</tr>
<tr>
<td></td>
<td>Example: &quot;<em>Test.doc&quot;</em></td>
</tr>
</tbody>
</table>
1.3 Used symbols

If safety instructions or other notes are used in this manual, they mean:

---

**Danger!**

The non-observance of this safety information may result in an imminent risk of death or severe injury:

- Observe the specified measures.

---

**Warning!**

The non-observance of this safety information may result in a potential risk of death or severe injury!

- Observe the specified measures.

---

**Caution!**

The non-observance of this safety information may result in a potential risk of injury or material damage!

- Observe the specified measures

---

**Note**

A note specifies special requirements or actions to be observed.

---

**Tip**

Tip or example as a helpful note or insider tip to make the work a little bit easier.

---

**Other documentation**

Reference to additional documentation or further reading.
2 Introduction

Different types of data stores are available in ibaPDA for different purposes and methods of data storage. Depending on the licenses registered in the dongle, different types of data stores are available for configuration in the dialog.

This documentation describes the “Kafka cluster time-based data store” type of recording. This type of data store writes time-based data into a Kafka cluster.

The architecture of Apache Kafka consists of a cluster computer network. So-called ‘brokers’ in this computer network store received messages with a time stamp in so-called ‘topics.’

Topics can be split into several partitions in the cluster and replicated. Applications that write messages in a Kafka cluster are called producers. The messages are sent to the leader. Among themselves the brokers take care of the replication with the followers. Consumers are applications that read messages from the Kafka cluster.

ibaPDA has the role of a producer in a Kafka environment: The data store sends messages to the cluster.

Additional information on Apache Kafka can be found at

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

Chapter 📖 Signal selection, page 16 describes the selection of the signals that are to be recorded in the topics.

The data can be continuously recorded or recorded by trigger. See chapter 📖 Trigger mode, page 18
2.1 System requirements

The following system requirements are necessary when using data storage in an Apache Kafka cluster:

- *ibaPDA v7.1.0 or higher*
- *License for ibiaPDA-Data-Store-Kafka*

The licenses are staggered according to the number of signals that should be written in the Kafka cluster. The number of used data stores used is unlimited.

<table>
<thead>
<tr>
<th>Order no.</th>
<th>Product name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.670160</td>
<td>ibiaPDA-Data-Store-Kafka-16</td>
<td>Additional license for data streaming into a Kafka cluster, max. 16 signals</td>
</tr>
<tr>
<td>30.670161</td>
<td>ibiaPDA-Data-Store-Kafka-64</td>
<td>Additional license for data streaming into a Kafka cluster, max. 64 signals</td>
</tr>
<tr>
<td>30.670162</td>
<td>ibiaPDA-Data-Store-Kafka-256</td>
<td>Additional license for data streaming into a Kafka cluster, max. 256 signals</td>
</tr>
<tr>
<td>30.670163</td>
<td>ibiaPDA-Data-Store-Kafka-1024</td>
<td>Additional license for data streaming into a Kafka cluster, max. 1024 signals</td>
</tr>
<tr>
<td>30.670171</td>
<td>upgrade-ibiaPDA-Data-Store-Kafka-16 to 64</td>
<td>License for extension from 16 to 64 signals</td>
</tr>
<tr>
<td>30.670172</td>
<td>upgrade-ibiaPDA-Data-Store-Kafka-64 to 256</td>
<td>License for extension from 64 to 256 signals</td>
</tr>
<tr>
<td>30.670173</td>
<td>upgrade-ibiaPDA-Data-Store-Kafka-256 to 1024</td>
<td>License for extension from 256 to 1024 signals</td>
</tr>
</tbody>
</table>

Table 1: Available licenses for the Kafka cluster data storage
3 Data store configuration

3.1 Add a data store

The dialog for data storage configuration can be opened in the Configure – Data storage main menu or by clicking on the button in the main toolbar.

In order to add a new data store, click on the blue link Add data store in the tree structure. You can also right-click on the data store node in the tree structure and choose Add data store from the context menu.

Select Kafka cluster timebased data store for the recording of timebased data in an Apache Kafka cluster.

![Add data store](image)

Fig. 2: Add a data store
3.2 Kafka cluster data store

Fig. 3: Configuring the Kafka cluster data store

Locked
A data store can be locked in order to prevent an accidental or unauthorized change of settings.

Active
A data store must be enabled in order to work. However, you can configure various data stores and disable data stores that are not required.

Data store name
You can enter a name for the data store here.

Maximum buffer size
The maximum buffer size determines how much data is buffered in ibaPDA in the event of a temporary connection loss with the Kafka cluster.

Identifier
The identifier is a text-based value that can be included in the datasets written in the cluster. For the subsequent processing of data, this may be useful for distinguishing between several ibaPDA systems that write into the same cluster.
Connection

Library Version
Display of the library version contained in ibaPDA (only for information). ibaPDA uses the open-source library librdkafka.

Type
There are two types to choose from

- Kafka: Generic Kafka cluster
- Event hub: for the connection with a Microsoft Azure event hub with enabled Kafka support

The following settings are required for the Kafka type:

Cluster address
Enter the host name and port of the broker in the Kafka cluster here. In case multiple brokers are to be addressed at the same time, separate the single entries by commas (e.g. 'hostname1:9092, hostname2:9092, hostname3:9092').

Schema registry address
If a schema registry is used in the Kafka cluster, then enter the host name and port of the schema registry here. If no schema registry is used, the field must remain empty.

Message timeout
Time in seconds that ibaPDA waits for a response from the Kafka cluster. The value corresponds to the parameter message.timeout.ms in the library librdkafka.

Acknowledgment mode

- None: Data messages are sent as a continuous stream, but without confirmation of whether they were received by the leader or not. This is the fastest mode, but it is not guaranteed that the data was acquired by a broker.
- Leader: ibaPDA waits until the topic leader provides the confirmation that the sent data has been acquired. If the leader malfunctions after sending the confirmation and before the followers have replicated the dataset, the data is lost.
- All: No new data will be sent from ibaPDA until the topic leader and the followers have confirmed that the data has been written in their records.

The value corresponds to the parameter acks in the library librdkafka.

Message batching time
Time in milliseconds that ibaPDA waits to send messages. If the value is 0, messages are sent to the Kafka cluster as quickly as possible. If the value is set to 100 ms, for example, ibaPDA sends packets with buffered messages every 100 ms. This increases the latency time, but reduces the processing effort both on the ibaPDA- and on the cluster side. The value corresponds to the parameter linger.ms in the library librdkafka.

In the bottom part of the dialog, you can configure additional parameters of the librdkafka library.
Data store configuration

Parameters that have already been created are listed in the list with the name and value. Next to the list, there are buttons with the following features:

<table>
<thead>
<tr>
<th>Button</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Add" /></td>
<td>Add parameters</td>
</tr>
<tr>
<td><img src="image" alt="Remove" /></td>
<td>Delete selected parameters</td>
</tr>
<tr>
<td><img src="image" alt="Import" /></td>
<td>Import parameters</td>
</tr>
</tbody>
</table>

Add parameters
Enter the name and a value in the fields Parameter and Value.

Delete selected parameters

Import parameters
You can import parameters as CSV files by selecting the CSV file in the file browser.

---

**Note**

Configured parameters in the list always have priority and overrule other settings.

The documentation of the parameters available in the *librdkafka* library can be found at:

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

---

The following settings are required for the **Event Hub** type:

![Connection settings](image)

**Connection string**
You can find the connection string in the properties of the event hub in the Microsoft Azure portal.

**Message timeout**
Time in seconds that *ibaPDA* waits for a response from the event hub.

Use the <Test connection> button to test the connection to the event hub.
3.3 Configuring topics

Several topics are usually defined in the Kafka cluster, which are created by the cluster administrator. It is not possible to create or delete topics in the cluster from ibaPDA.

If you highlight the Topics node, the list of topics is shown, that are to be written by the data store. If it is possible to connect with the cluster, you can use the button to download the list of the topics defined in the cluster. If the connection to the cluster is missing, in order to prepare a configuration of the data store, it is also possible to use the button to add topics to the list. Use to delete topics from the configuration.

Buttons for the configuration of the topics:

- Download the topic list from the Kafka cluster (a connection to the cluster is required)
- Manually add a new topic
- Remove selected topic
- Copy selected cell content to the clipboard
- Paste data from the clipboard into cells

**Note**

Use <Shift> and <Ctrl> to select multiple cells and copy/paste/delete them.

Meaning of the columns:
Active
Writing in a topic can be enabled/disabled by enabling the checkbox.

Name
A topic is identified by its name.

Key
The Key property is optional. The property can be defined by a series of placeholders:

- $identifier: Identifier, defined in the data store configuration, see chapter Kafka cluster data store, page 10
- $signalid: Signal ID
- $signalname: Signal name
- $unit: Signal unit
- $comment1: Signal comment 1
- $comment2: Signal comment 2

When using the grouped data format JSON (grouped), signal-related placeholders are replaced by empty text.

Metadata
The following metadata can be inserted in the telegram:

Signal unit, comment 1/2, current timestamp, signal name, signal ID, the identifier configured in the Kafka data store configuration. Select the desired metadata with a check mark.

Note
The metadata is included in every message. Since all metadata is constant during a measurement, except for the timestamp, this leads to a lot of redundant data being written in the cluster. It is therefore important to consider which metadata is actually required. The timestamp is implicitly already included in every message. Adding the timestamp to the metadata of a message therefore also leads to redundant information. From the data analysis perspective, however, it may be useful to enable the timestamp in the metadata.

Signal reference
Use the signal reference to define whether the signal ID, the signal name or a custom text should be used in a dataset in order to distinguish between several signals. The same placeholders as for the key column can be used for the custom text.

Data format
The following data formats are supported:
**JSON (grouped):** all signal data for a certain timestamp is combined in a message

**JSON (per signal):** one message is created per signal and timestamp

**AVRO (per signal):** one message is created per signal and timestamp

You will find examples of the description of the data formats in chapter "Description of the data formats," page 24.

**Signals**

In order to configure the signals that are to be written in a topic, select the topic in the structure tree or click on the <...> button in the Signals column.

![Kafka cluster timebased data store 1 - Topics](image)

**Fig. 6: Calling up the signal selection**

In the following dialog, you assign the desired signals to the topics using the storage profiles. See chapter "Signal selection," page 16.
4 Signal selection

To enable signals to be recorded, they must be assigned to a topic by using a storage profile of the Time type. Click in the Topics node, in the Signals column of the topic list on the <...> button to access the signal dialog.

![Kafka cluster timebased data store 1 - Topics](image)

Fig. 7: Calling up the signal selection

Or select a topic in the tree structure.

![Signal selection dialog](image)

Fig. 8: Signal selection dialog

**Note**

Additional information about the storage profiles can be found in the *ibaPDA* manual, part 5.

Select the topic to which you would like to assign certain signals and select a storage profile in the profile list. Set a check mark in the selection fields next to the signals which you would like to assign to this profile.
Kafka cluster data stores are licensed per number of written signals. The current number of selected signals in all Kafka cluster data stores is shown at the bottom of the dialog, similar to the number of configured signals in the I/O Manager. The length of the bar corresponds to the licensed number of signals. In the example above, a maximum of 1024 signals can be written via Kafka data stores. 4 signals are currently already used.
5 Trigger mode

The description applies to the following types of data stores:

- Database/cloud timebased
- Kafka cluster timebased
- Mindsphere timebased
- MQTT timebased.

In the Trigger Mode node, you determine when data is recorded.

Start trigger

You initially choose whether you would like to continuously record or it should be fired by a trigger.

Unconditional

The data is continuously recorded with this selection. In this case, the recording will start immediately at the start of the measurement or when pressing the "GO" button.

Trigger on signal

If you want the trigger to fire on a measured signal or a virtual signal, you need to check Trigger on signal in the option field. In the fields next to this, define the properties of the trigger signal.

- Field 1: Drop-down list for signal selection (available analog and digital signals)
- Field 2: Drop-down list for selecting edges or levels
Field 3: Drop-down list for selecting the trigger level value given in the specific physical unit (field 3 is only enabled in case of analog trigger signals)

Both analog and digital signals can serve as triggers. The signal to trigger on is to be selected from the drop-down lists (see picture below, field 1). In the drop-down list, you will find the well-known signal tree containing available signals. Select the signal you want to use as trigger signal.

![Fig. 10: Configuring "Trigger on signal"](image)

Depending on whether a digital or an analog signal was selected, the fields 2 or 3, respectively, are offered allowing the trigger event to be defined more specifically.

As for analog signals, you can choose between level or edge triggers including a predefined level (field 3).

![Fig. 11: Configuring "Trigger on signal", analog signal, edge or level](image)

As for digital signals, you can choose between level or edge triggers including the 2 levels logical 0 (FALSE) and logical 1 (TRUE).

![Fig. 12: Configuring "Trigger on signal", digital signal](image)
Trigger mode

If you want to use a start trigger always at a certain time regularly, you can check the “Trigger every ... minutes starting at ...” option. Enter the period given in minutes, or select it from the input field. Value range is from 0 to 1440, which equals one day. Then enter or select the start time for the first trigger. Value range is from 00:00 to 23:59, which equals one day.

1 sample on change
A new sample will be taken each time the value of the trigger signal changes. The deadtime determines the minimum amount of time between samples.

Pre-trigger time
You can configure a pre-trigger time and then the recording begins by the pre-trigger time before the trigger event. If the trigger condition is met, the incoming data is added to the data buffered during the pre-trigger time.

Trigger dead time
This property is available for the start triggers “Trigger on signal” and “Trigger every ...”. The trigger dead time determines the time of suppressing subsequent triggers after a trigger occurred.

If the dead time, for instance, is set to 5 seconds, all other triggers are ignored for the duration of 5 seconds after the first trigger occurrence.

Trigger at the start of the acquisition
If you want the recording to start immediately at acquisition start or as soon as you apply a new data storage configuration, you also need to check the Trigger on acquisition start option. If you do not enable this option, the recording first starts once the trigger is fired.

If start trigger occurs again while file is already recording, then:
You can determine here what should happen if a new start trigger occurs while a recording is already running.

- Ignore it:
  Selecting this option will cause the system to ignore any new start trigger during a running recording for as long as the stop trigger occurs

- Extend recording time:
  If this option is enabled, it extends the duration of the running recording upon occurrence of another start trigger during an ongoing recording. This occurs as often as set in the "Maximum number of extensions on single file" field. If the max. number of extensions is reached, all subsequent start triggers will be ignored. Of course, the recording is stopped immediately by any stop trigger.

Stop trigger
The settings for the stop trigger are made in the same way as those for the start trigger. Here, both analog and digital signals can also be used as triggers.

Trigger after recording of x hours x minutes x seconds
Here you can configure a time span according to which the recording is ended - after the occurrence of the start trigger.
**Trigger on signal**
See explanation for start trigger above.

**Post trigger time**
You can configure a post trigger time and then the recording ends by the post trigger time after the stop trigger event.
6 Diagnostics

6.1 Data storage status

The data storage status window shows the current status of the data stores.

![Data storage status window](image)

Fig. 13: Example of data storage status window

All defined data stores and their respective status are displayed here, depending on the data store, with server address, acquisition duration, write speed, etc.

The icon in front of the name indicates the current status of the storage:

- **T**: Wait for the start trigger (only for triggered recording)

- **Recording in progress**

- **P**: Post-trigger phase; stop trigger occurred, but acquisition continues until the post-trigger time is over

  Disabled or faulty data store is indicated by a red cross in the data store icon.

Right-clicking on this node allows you to manually send a start or stop trigger.
6.2 Diagnostics of data stores

The Diagnostics node in the data storage tree offers information about the system load by the data stores. The measurement must be running.

![Fig. 14: Diagnostics of data stores](image)

The performance values of all data stores are shown in the table. There is one row per data store. The rows are grouped according to the threads that write the data.

In each group row is the name of the thread and (in brackets) its share of the load. The load average is displayed by default. But, you can switch between the average and actual value using the context menu.

The Disk column indicates the respective target to which the data is written, for example a hard disk partition, the address of the database, the address of the Kafka cluster, etc. The Write speed indicates how fast the data is written. The Buffered data columns indicate how much data is buffered in ibaPDA.

The Acquisition Thread Load column indicates various information depending on the data stores. For timebased data stores, the Acquisition Thread Load column indicates the amount of time needed for the run length encoding and writing to a disk or in a Kafka cluster. For database/cloud data stores, the column indicates the load caused by the analysis of the triggers and creation of the row data.

Additional information about diagnostics can be found in the ibaPDA manual, part 5.
7 Appendix

7.1 Description of the data formats

7.1.1 Data format JSON (grouped)

The following explains the setup of the data format “JSON (grouped)” using an example with three signals. The signals are defined as follows:

```
json example for signal ID as a signal reference
{

  "[0:0]": Actual value,
  "[0:0].ID": "[0:0]",
  "[0:0].Name": "Signalname_0",
  "[0:0].Unit": "Unit_0",
  "[0:0].Comment1": "Example_comment1_0",
  "[0:0].Comment2": "Example_comment2_0",
  "[0:1]": Actual value,
  "[0:1].ID": "[0:1]",
  "[0:1].Name": "Signalname_1",
  "[0:1].Unit": "Unit_1",
  "[0:1].Comment1": "Example_comment1_1",
  "[0:1].Comment2": "Example_comment2_1",
  "[0:2]": Actual value,
  "[0:2].ID": "[0:2]",
  "[0:2].Name": "Signalname_2",
  "[0:2].Unit": "Unit_2",
  "[0:2].Comment1": "Example_comment1_2",
  "[0:2].Comment2": "Example_comment2_2",
  "Timestamp": "2020-01-21T13:10:53.0002189Z",
  "Identifier": "My identifier"
}
```

Red: optional signal-related metadata
Green: optional group-related metadata
JSON example for signal name as a signal reference

```json
{
    "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",
    "Identifier": "My identifier"
}
```

Red: optional signal-related metadata
Green: optional group-related metadata
7.1.2 Data format JSON (per signal)

The following explains the setup of the data format “JSON (per signal)” using an example with three signals. The signals are defined as follows:

![Fig. 16: Signals in the analog tab](image-url)
**JSON example for signal ID as a signal reference**

```json
[
  "Signal": "[0:0]",
  "Value": "Actual value",
  "ID": "[0:0]",
  "Name": "Signalname_0",
  "Unit": "Unit_0",
  "Comment1": "Example_comment1_0",
  "Comment2": "Example_comment2_0",
  "Timestamp": "2020-01-21T13:26:50.87840742",
  "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": "2020-01-21T13:26:50.87840742",
  "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": "2020-01-21T13:26:50.87840742",
  "Identifier": "My identifier"
]

Red: optional signal-related metadata
JSON example for signal name as a signal reference

```json
{
  "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"
}

Red: optional signal-related metadata
7.1.3 Data format AVRO (per signal)

Compared to JSON, the data is saved in a readable format. AVRO uses a binary coding, which reduces the bandwidth and the required disk space.

*ibaPDA* uses the following schema for the serialization of the signal data:

```json
{
   "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", "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* also supports a connection to a confluent schema registry. However, this is not necessarily required so that the field can be left empty. If a schema registry is configured, *ibaPDA* registers the schema used for the coding in the schema registry. The returned ID is then appended to each dataset as follows:
The address and port of the schema registry can be entered in the schema registry address field. Use the <Test connection> button to test the connection to the schema registry. If no schema registry is used, each dataset will be coded as follows:

<table>
<thead>
<tr>
<th>Byte offset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0x00 (Confluent AVRO Marker)</td>
</tr>
<tr>
<td>1</td>
<td>Schema ID (Big endian)</td>
</tr>
<tr>
<td>5</td>
<td>Signal data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Byte offset</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0xC3 0x01 (Single-object encoding marker)</td>
</tr>
<tr>
<td>2</td>
<td>CRC-64-AVRO fingerprint of encoding schema</td>
</tr>
<tr>
<td>10</td>
<td>Signal data</td>
</tr>
</tbody>
</table>

**Note**

You can find detailed information about Avro here:

8 Support and contact

Support

Phone: +49 911 97282-14
Fax: +49 911 97282-33
Email: support@iba-ag.com

Note

If you require support, indicate the serial number (iba-S/N) of the product or the license number.

Contact

Headquarters
iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Phone: +49 911 97282-0
Fax: +49 911 97282-33
Email: iba@iba-ag.com

Mailing address
iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address
iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and Worldwide
For contact data of your regional iba office or representative please refer to our web site