



# ACS

## ACS Kafka Integration

From version 2.11.5

Copyright © 2025, Ubisense Limited 2014 - 2025. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Ubisense at the following address:

Ubisense Limited  
St Andrew's House  
St Andrew's Road  
Cambridge CB4 1DL  
United Kingdom

Tel: +44 (0)1223 535170

WWW: <https://www.ubisense.com>

All contents of this document are subject to change without notice and do not represent a commitment on the part of Ubisense. Reasonable effort is made to ensure the accuracy of the information contained in the document. However, due to on-going product improvements and revisions, Ubisense and its subsidiaries do not warrant the accuracy of this information and cannot accept responsibility for errors or omissions that may be contained in this document.

Information in this document is provided in connection with Ubisense products. No license, express or implied to any intellectual property rights is granted by this document.

Ubisense encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.

UBISENSE®, the Ubisense motif, SmartSpace® and AngleID® are registered trademarks of Ubisense Ltd. DIMENSION4™ and UB-Tag™ are trademarks of Ubisense Ltd.

Windows® is a registered trademark of Microsoft Corporation in the United States and/or other countries. The other names of actual companies and products mentioned herein are the trademarks of their respective owners.

# Contents

---

<b>ACS Kafka Integration</b> .....	<b>1</b>
Overview .....	1
Getting started .....	1
Installation .....	1
Kafka protocol .....	1
Configuration .....	1
External systems .....	1
Associated Events .....	3
Operation .....	4
Connecting to the Kafka cluster .....	4
Live logs .....	5
Kafka messages .....	5

# ACS Kafka Integration

---

## Overview

From version 2.11.5, ACS includes a **Kafka ACS integration** installable feature which lets you configure ACS events to be sent to a Kafka cluster via the service **ACS Kafka events**. Support is built on existing ACS tools for configuring external systems.

## Getting started

### Installation

Install the **Kafka ACS integration** feature from the release packages using Service Manager.

See [Kafka quickstart](#) if you do not have a Kafka cluster.

#### **Additional requirements for Linux installations**

You must install **librdkafka** so that services can find **librdkafka.so.1**.

#### **Additional requirements for Windows installations**

You must install OpenSSL 3.x libraries and add them to your system PATH. For example, when downloading the libraries from [FireDaemon](#), use the MSI installer and tick the **Adjust PATH system environment** box.

### Kafka protocol

Once the service **ACS Kafka events** is running, the **Kafka** protocol is listed under **ACS > Configuration > External Systems > Protocols** in the ACS Main GUI.

## Configuration

### External systems

Create an external system for each Kafka cluster you want to send ACS events to.

Parameter	Meaning
Name	Unique name of the external system, which you can choose according to your naming conventions
Type	Unused by the Kafka protocol
Description	Optional human-readable description
IP Address	The value combined with the port number and passed to the Kafka <b>bootstrap.servers</b> property
Port	The value appended to the IP address and passed to the Kafka <b>bootstrap.servers</b> property
Protocol	Must be "Kafka" for external systems corresponding to a Kafka cluster
Protocol Version	Must be "1.0" for this version
Remote station is Server	Unused by the Kafka protocol
Accepted IP Addresses	Unused by the Kafka protocol

There are two additional parameter requirements that you can set for each external system using the Kafka protocol. If your cluster does not have security and you want to use only default values for the Kafka producer configuration, just set the IP address and port to point to the cluster, and the values are passed to the Kafka **bootstrap.servers** property. For all other cases, set **Producer settings** to point to a file on the server containing key-value pairs for any keys described in [producer-configs](#).

Additional parameter	Meaning	Default value
Producer settings	Path to a file containing key-value pairs of producer configuration	<empty>
Topic	Kafka topic to use for all events associated with the external system	ACS-Events
Key	Key to use for Kafka messages (default is to use JSON with the event and activator)	<empty>

If the producer settings file contains a value for **bootstrap.servers**, it takes precedence over the IP address and port configured in the UI.

The topic is per-external system, so if you want to split ACS events around different Kafka topics, use multiple external systems as required.

### Example producer config

```
# list of brokers used for bootstrapping knowledge about the rest of the cluster
# format: host1:port1,host2:port2 ...
bootstrap.servers=localhost:9092

# specify the compression codec for all data generated: none, gzip, snappy, lz4, zstd
compression.type=none

# name of the partitioner class for partitioning records;
# The default uses "sticky" partitioning logic which spreads the load evenly between
    ↪ partitions, but improves throughput by attempting to fill the batches sent to
each
    ↪ partition.
#partitioner.class=

# the maximum amount of time the client will wait for the response of a request
#request.timeout.ms=

# how long `KafkaProducer.send` and `KafkaProducer.partitionsFor` will block for
#max.block.ms=

# the producer will wait for up to the given delay to allow other records to be sent
so
    ↪ that the sends can be batched together
#linger.ms=

# the maximum size of a request in bytes
#max.request.size=

# the default batch size in bytes when batching multiple records sent to a partition
#batch.size=

# the total bytes of memory the producer can use to buffer records waiting to be sent
to
    ↪ the server
#buffer.memory=
```

## Associated Events

Once you have set up your external system with the "Kafka" protocol, configuring named events to be sent to the Kafka cluster is straightforward: just tick the external system in the list of external systems associated with the event. For example, in the ACS Main GUI, lists with checkboxes for each external system appear in:

- ACS > Configuration > Products > ProductsIsLocated Events
- ACS > Configuration > Factory Layout > Workspaces > Events

- ACS > Configuration > Factory Layout > Assembly Line Stations > Events
- ACS > Configuration > Factory Layout > Trigger Points > Events
- ACS > Configuration > Factory Layout > Ident Zones > Events
- ACS > Configuration > Devices > Device Instances > Events

You can see an overview of all associated events in ACS > Configuration > External Systems > External Systems > Associated Objects. You can add or remove events to send to the Kafka cluster here as well.

Although you can associate association zones, disassociation zones, and assembly lines with any external system, these will be ignored by those configured with the Kafka protocol, which is only concerned with events.

### Outgoing ACS Events

Since Outgoing ACS Event objects configured in SmartSpace do not have their own dedicated page in the ACS Main GUI, they can only be associated with the external system via its list of associated objects.

## Operation

### Connecting to the Kafka cluster

In the ACS Main GUI, use ACS > Operation > Connection status to connect or disconnect the Kafka external systems.

You can also set the **Requested Connection Status** property using SmartSpace Config or Business rules as required.

In this context, "**connection**" refers to whether the ACS Kafka Events service creates a Kafka producer corresponding to the configured external system.

Connection status is one of the following:

Status	Meaning
Disconnected	No Kafka producer instantiated for this external system
Connecting	Kafka producer instantiated; connection messages appear in the <b>Info</b> column
Connected	Kafka producer instantiated and most recent message was sent successfully

If there are no associated events, or the associated events occur infrequently, the status will be "Connecting" with info "Waiting for an event to send" until an attempt is made to send a message to the Kafka cluster.

## Live logs

In the ACS Main GUI, use **ACS > Operation > Live Logs > External System Messages** to see what the service is doing for each external system, including what events are going to the Kafka cluster.

You can also enable the trace stream "acs\_kafka" for `ubisense_trace_receiver` if required.

## Kafka messages

When an event occurs, each connected external system checks whether it is associated with the event, and if so, sends it to the Kafka cluster, giving something similar to the following:

```
Topics : ACS-Events
Key    : {
      "activator": "car 00001",
      "event": "product is located"
    }
Value  : {
      "CurrentSpaces": [
        "workspace"
      ],
      "InsertedSpaces": [
        "workspace"
      ],
      "ProductName": "car 00001",
      "ProductType": "Car",
      "RemovedSpaces": [],
      "activator": "car 00001",
      "assertionTime": "1735835496.0933502",
      "eventClass": "ProductIsLocatedEvent",
      "event": "product is located",
      "timeout": 18446744073.709553
    }
```



The value fields depend on the kind of event: this example is a ProductIsLocatedEvent.

### Key

By default, the key is a JSON object that includes the name of the Ubisense object that triggered the event (the "activator"), and the name of the event.

You can override this behavior using the **Key** external system parameter if required.



**Value**

Field	Meaning
activator	Human-readable name of the activator object in the key
assertionTime	The time the event happened, in number of seconds UTC since 00:00:00, 1st January 1970
eventClass	The kind of event, for example ProductIsLocatedEvent
event	Human-readable name of the event object in the key
timeout	The time when the event times out, in number of seconds UTC 00:00:00, 1st January 1970

The remaining fields in the message value are specific to the various event classes. Event classes include:

- ContainerContainsObjectEvent
- ProductHasPassedTriggerPointEvent
- ProductIsLocatedEvent
- ObjectIsLocatedEvent
- TagIsLocatedEvent
- Outgoing ACS Event (or, more usually, types configured in SmartSpace that inherit Outgoing ACS Event)

Depending on which custom services are installed in the dataset, there may be more event classes available. These appear in the dropdown when adding rows to the associated object table in the UI.