Kafka

The **Kafka** communication point enables Rhapsody to publish (write) or subscribe to (read) streams of events, respectively, from a Kafka streaming platform (a cluster of Kafka brokers).

> **Note**
>
> For details on Kafka and Kafka terminology, refer to the Kafka Documentation[https://kafka.apache.org/documentation/] website.

Supported Operational Modes: *All*

Design Considerations

The Kafka communication point is a versatile Rhapsody component and can be used in a variety of solutions.

With an output Kafka communication point, you can use Rhapsody to convert your system into a Kafka producer for publishing events to a topic in a Kafka cluster. For example, Rhapsody can enable a Patient Administration System to publish data to a Kafka-powered dashboard for monitoring purposes.

With an input Kafka communication point, you can use Rhapsody to convert your system into a Kafka consumer for subscribing to events from topics in a Kafka cluster. For example, Rhapsody can be deployed to populate a clinical data repository with data from a Kafka cluster.

When designing end-to-end Kafka-based systems, you can insert Rhapsody between a Kafka producer and a Kafka cluster for pre-processing event streams into a form suitable for downstream Kafka consumers. Using Rhapsody in this way provides the advantage of simplifying and future-proofing the addition of Kafka consumers, and compensating for any changes to the input stream in a single place, without impacting multiple downstream systems.

Within a Kafka broker, cluster, or group of clusters, Rhapsody can be used to process and send events directly from one set of topics to another. Again, in this scenario, Rhapsody can pre-process events into a form suitable for downstream Kafka consumers or, more generally, provide an easy-to-configure replacement for inter-topic Kafka business logic.

The Kafka streaming platform is designed to handle high rates of data transfer. If you require comparable levels of performance from your Rhapsody interface, test your configuration in a test environment in order to ensure the rate of the consumed stream matches Rhapsody's

capabilities before promoting it to production. In cases where your Rhapsody instance is approaching its performance limits, consider spreading the load across multiple Rhapsody instances or using Kafka topics to filter events.

Operational Modes

The Kafka communication point can operate in the following modes:

- *Input* - subscribes to events from Kafka topics.
- *Output* - publishes events to a single Kafka topic.
- *Bidirectional* - publishes and subscribes to events from a Kafka topic.
- *In->Out* - subscribes to and publishes events from topics in a synchronized manner.
- *Out->In* - publishes and subscribes to events from a Kafka topic in a synchronized manner.

Input Mode Properties

Property	Description
Bootstrap Server URLs	The list of Kafka brokers to use for establishing the initial connection to the Kafka cluster. Each broker entry has host and port values.
SASL Authentication Mechanism	The SASL authentication mechanism to use for client authentication with Kafka brokers: • `NONE` (default) - SASL authentication is disabled. • `PLAIN`. • `SCRAM-SHA-256`. • `SCRAM-SHA-512`. SSL client authentication is automatically disabled when SASL authentication is enabled (in other words, when `PLAIN` or one of the SCRAM options is selected).
SASL Username	The username for the authenticated principal used for SASL authorization. Required when SASL authentication is enabled.
SASL Password	The password for the authenticated principal used for SASL authorization. Required when SASL authentication is enabled.
Subscribed Topic(s)	The list of Kafka topics from which to subscribe to events.

Property	Description
Use SSL	Whether to use SSL: • Enabled - secure socket layer is used for this connection and all events are encrypted. • Disabled (default). **Note** It is recommended that you enable SSL for the Kafka communication point whenever SSL is supported by the remote server. Refer to TLS/SSL Support in Rhapsody[tls-ssl-support-in-rhapsody.html] for details. The Kafka communication point only supports TLS v1.2.
SSL Cipher Suites	The cipher suite to use when creating an SSL connection. Refer to SSL Cipher Suites[tls-ssl-support-in-rhapsody.html#ssl-tls-cipher-suites] for details.
Trusted Certificates	Identifies the trusted certificates used at the non-Rhapsody end of the SSL connection. If it is not configured with the SSL server's certificate, then this SSL client operates in anonymous server authentication mode. It is highly recommended that you always provide the server's certificate here to allow server authentication to be performed. Refer to Server Authentication[tls-ssl-support-in-rhapsody.html#server-authentication] for details.
Require Client Authentication	Whether to require client authentication from the Kafka broker: • Enabled - configures the Kafka broker to request client authentication. • Disabled (default).
Secure Keys	Identifies the private keys used for authenticating Rhapsody's end of the SSL connection. When configured with a private key, the SSL client supports SSL client authentication if requested by the SSL server. Refer to Client Authentication[tls-ssl-support-in-rhapsody.html#client-authentication] for details.

Property	Description
Additional Security Properties	A list of additional name-value pairs of Kafka client configuration security properties to be set on request messages in order to fine-tune security settings. Kafka client properties whose names start with `ssl.` or `sasl.` can be configured via this property. For example: • `ssl.endpoint.identification.algorithm` - the endpoint identification algorithm used by clients to validate the server. • `hostname.sasl.login.connect.timeout.ms` - the (optional) value in milliseconds for the external authentication provider connection timeout. The property name is required, but the property value can be null.
Group ID	A unique string that identifies the consumer group the communication point belongs to. **Note** If an input mode Kafka communication point is stopped and started with the same Group ID, it continues to read events from where it left off; if it is stopped and started with a new Group ID, then it reads events from the initial offset maintained by the Kafka broker.

Output Mode Properties

Property	Description
Bootstrap Server URLs	The list of brokers to use for establishing the initial connection to the Kafka cluster. Each broker entry has host and port values.
SASL Authentication Mechanism	The SASL authentication mechanism to use for client authentication with Kafka brokers: • `NONE` (default) - SASL authentication is disabled. • `PLAIN`. • `SCRAM-SHA-256`. • `SCRAM-SHA-512`. SSL client authentication is automatically disabled when SASL authentication is enabled (in other words, when `PLAIN` or one of the SCRAM options is selected).

Property	Description
SASL Username	The username for the authenticated principal used for SASL authorization. Required when SASL authentication is enabled.
SASL Password	The password for the authenticated principal used for SASL authorization. Required when SASL authentication is enabled.
Use SSL	Whether to use SSL: • Enabled - secure socket layer is used for this connection and all events are encrypted. • Disabled (default). **Note** It is recommended that you enable SSL for the Kafka communication point whenever SSL is supported by the remote server. Refer to TLS/SSL Support in Rhapsody[tls-ssl-support-in-rhapsody.html] for details. The Kafka communication point only supports TLS v1.2.
SSL Cipher Suites	The cipher suite to use when creating an SSL connection. Refer to SSL Cipher Suites[tls-ssl-support-in-rhapsody.html#ssl-tls-cipher-suites] for details.
Trusted Certificates	Identifies the trusted certificates used at the non-Rhapsody end of the SSL connection. If it is not configured with the SSL server's certificate, then this SSL client operates in anonymous server authentication mode. It is highly recommended that you always provide the server's certificate here to allow server authentication to be performed. Refer to Server Authentication[tls-ssl-support-in-rhapsody.html#server-authentication] for details.
Require Client Authentication	Whether to require client authentication from the Kafka broker: • Enabled - configures the Kafka broker to request client authentication. • Disabled (default).

Property	Description
Secure Keys	Identifies the private keys used for authenticating Rhapsody's end of the SSL connection. When configured with a private key, the SSL client supports SSL client authentication if requested by the SSL server. Refer to Client Authentication[tls-ssl-support-in-rhapsody.html#client-authentication] for details.
Additional Security Properties	A list of additional name-value pairs of Kafka client configuration security properties to be set on request messages in order to fine-tune security settings. Kafka client properties whose names start with `ssl.` or `sasl.` can be configured via this property. For example: • `ssl.endpoint.identification.algorithm` - the endpoint identification algorithm used by clients to validate the server. • `hostname.sasl.login.connect.timeout.ms` - the (optional) value in milliseconds for the external authentication provider connection timeout. The property name is required, but the property value can be null.
Topic	The Kafka topic to publish events to. The communication point can only publish events to a single topic. **Note** If the specified topic does not exist in the Kafka broker, a new topic with the provided name is created. The topic can be retrieved from a message property by using the `$MessageProperty` notation. For example, if **Topic** is configured to `$kafkaTopic`, the value of the message property `kafkaTopic` is used as the topic for that message.
Partition	The partition number of the Kafka topic to publish events to.
Key	The key to be included in events during publishing. The key can be retrieved from a message property by using the '`$MessageProperty`' notation. For example, if **Key** is configured to `$kafkaKey`, the value of the message property `kafkaKey` is used as the key for that message.
Headers	The headers to be included in events during publishing.

Bidirectional, In->Out and Out->In Mode Properties

The configuration properties for this communication point in *Bidirectional*, *In->Out* and *Out->In* modes are a combination of the properties described above.

Refer to Out->In and In->Out Properties[connection.html#in--out-and-out--in-properties] for general details on a communication point's *In->Out* and *Out->In* modes.

Logging for Kafka Streaming

The `logback.xml` file includes a log appender specifically for logging Kafka information. Kafka logs can be used for troubleshooting problems with establishing a connection or publishing and subscribing to events. Refer to Kafka Logging[default-logging-in-rhapsody.html#kafka-logging] for details.

 Last modified: 2025-08-20