

Configuring Kafka Producers and Subscribers

Last Modified on 04/07/2020 7:08 am EDT

V9.2.2 - 9.3

Overview

You configure Kafka activities to set up a messaging mechanism within Cora SeQuence or between Cora SeQuence and other applications. Apache Kafka® is a distributed streaming platform that is designed to be fast, scalable, and durable. Kafka is generally used to move data between systems or applications and to enable applications to consume data required to perform specific actions.

Use cases

Kafka messaging mechanism can be used in different scenarios. Following are a few examples:

- Asynchronous communication with external systems:
 - One-to-one communication scenario:
 1. Receive messages from an ERP system to initiate an invoice approval workflow.
 2. After invoice approval, send message back to the ERP system for further processing.
- Event publishing:
 - One-to-many communication scenario: Send messages to a topic that has multiple subscribers.
 1. Send a message to the topic after a payment process has completed.
 2. Multiple systems that subscribe to the topic perform an action based on the message.
- Multiple tasks per single message:
 - Handle a message sent from an external system to trigger multiple workflows:
 - A solution for welcoming a newly hired employee. One message with information about the new employee triggers multiple workflows, such as starting a workflow that requests the IT department to supply a laptop to the new employee, another workflow that requests the Security department to issue an ID tag, and yet another workflow for the HR department to add the employee to the relevant systems.

Configuration

Configuring a Kafka messaging on Cora SeQuence involves the following steps:

Steps	Performed by
1. Create the Kafka connection	Cora SeQuence Administrator
2. Add a Kafka producer	Cora SeQuence Administrator

NOTE: Required only for the Kafka Producer activity.

Steps	Performed by
3. Configure the Kafka integration activities in the workflow: <ol style="list-style-type: none"> a. Configure the Kafka Producer activity b. Configure the Kafka Subscriber activity 	Developer

For more details on how Apache Kafka works in Cora SeQUENCE, see [this article](#).

Create the Kafka connection

Configure the connection to the Kafka service. You can create new connections, edit existing ones, or delete them.

Prerequisites

Before you create a Kafka connection string, make sure that you have:

- A Kafka deployment (either SAAS or IAAS) is up and running.
- The Kafka connection details, including the required credentials.

For more information on the Kafka server requirements, see [this article](#).

Procedure

1. Go to **Administration > Global Settings > Kafka Connections**, and click **Add New Record**.
2. Enter a meaningful name for the Kafka connection string.
3. Enter the Connection URL.
4. If required, enter the relevant credentials to access the Kafka server.
5. By default, Cora SeQUENCE uses the parameters configured on the Kafka server. To change the default settings, click **Advanced Options**.

To learn more about Kafka parameters, refer to the [Kafka documentation](#).

Add a Kafka Producer

Kafka producer is the mechanism that publishes messages to one or more topics. You need to set up at least one Kafka Producer.

Prerequisite

- Make sure that a Kafka connection exists.

Procedure

1. Go to **Administration > Global Settings > Kafka Connections**, and click **Add New Record**.
2. Enter a meaningful name for the Kafka Producer.
3. Select the Kafka connection.
4. By default, Cora SeQUENCE uses the parameters configured on the Kafka server. To change the default settings, click **Advanced Options**.
5. Click **Add**.

Delivery guarantee

By default, the Kafka Producer does not acknowledge message delivery. When setting up Kafka to handle financial transactions, for example, you need to make sure that every message is delivered.

To set up delivery guarantee, configure the following parameters under **Advanced Options**.

Parameter	Description	Value
BatchNumMessages	Determines the number of messages per batch.	1
DeliveryReportFields	Determines the fields listed in the delivery reports. Supported values: key, value, timestamp, headers, all, none.	all
DeliveryReportTimeout	Determines the time, in milliseconds, to wait for a response from the Kafka server. When the configured period of time elapses, a failure message is logged.	3000
EnableDeliveryReport	Determines whether or not to enable notification of delivery reports.	True
LingerMs	Determines a delay, in milliseconds, before the Producer sends the message to the topic.	0.5

Advanced Options screen

Kafka Producer Advanced Options

Misc

BatchNumMessages: 1

CompressionLevel: []

CompressionType: []

DeliveryReportFields: all

DeliveryReportWaitTimeout: 3000

EnableBackgroundPoll: []

EnableDeliveryReports: True

EnableGaplessGuarantee: []

EnableIdempotence: []

LingerMs: 0.5

MessageSendMaxRetries: []

MessageTimeoutMs: []

Partitioner: []

QueueBufferingBackpressureThreshold: []

QueueBufferingMaxKbytes: []

QueueBufferingMaxMessages: []

RequestTimeoutMs: []

RetryBackoffMs: []

OK Cancel

When you set up Kafka to acknowledge delivery, the Producer sends one message at a time. After the message is successfully sent, the workflow continues, and the activity is marked as successful.

- If there are connection issues, a failure response is immediately sent to Cora SeQuence.
- If other issues occur, such as topic overflow, Kafka returns a failure status after a certain delay.

After the failure message is returned, Cora SeQuence sets the activity to failed.

NOTE

Carefully consider between the need for delivery guarantee and performance requirements. When you configure delivery guarantee, the system performs additional actions and activities that impact performance.

To learn more about Kafka parameters, refer to the [Kafka documentation](#).

Configure the Kafka Producer activity

The Kafka Producer *activity* is responsible for sending the message object and the topic name to the Kafka *Producer*.

Each activity instance can define a different message based on Cora SeQuence expressions.

Prerequisite

- A Kafka Producer has been configured.

Procedure

1. To add a Kafka Producer activity to your workflow, in the App Studio, select **Integration>Kafka Producer**.
2. On the Kafka Producer Properties screen, enter a significant name and alias, and then click **Next**.
3. Select a Kafka Producer from the list, or create a new one.
4. Click **Next**.
5. Click `ProducerRecord message`, and clear the **IsNull** option.
6. To show the configurable properties, expand the `ProducerRecord message` node.
 - Enter topic name.
If the topic does not exist, depending on Kafka's configuration, a topic can be automatically created.
 - Set the content and body of the message.
 - Set the key.
7. Click **Finish**.


Configure the Kafka Subscriber activity

The subscriber connects to Kafka and retrieves the message from a topic. Each subscriber connects to a specific topic. The Kafka Subscriber is a JES job.

Prerequisites

- A Kafka producer has been configured.
- Make sure that you have the relevant Group ID.
To learn more about groups in Kafka, refer to [this page](#).
- Obtain the relevant Kafka topic name from the Project Manager or Business Analyst.

Procedure

1. To add a Kafka Subscriber activity to your workflow, in the App Studio, select **Integration>Kafka Subscriber**.
2. Click the  Kafka Subscriber activity.
3. On the Kafka Subscriber Activity Properties screen, enter a significant name for the activity, and then click **Next**.
4. On the job tab, set the following:
 - **Name:** Enter a name for the actual job performed by the Kafka Subscriber activity.
 - **Scaling:** Set the required scaling settings. (Scaling is available only for **Cora SeSequence V9.3** and later versions.)
For more details, see [this article](#).
 - **Job is enabled:** Select this option only after you complete the workflow, or if you want to run the job for testing purposes.
5. Click **Next**.
6. On the Command tab, set the following:
 - **Kafka Connection** (mandatory): Select the relevant connection string.
 - **Group ID** (mandatory): Enter the group ID relevant to your implementation.
 - **Topic Name** (mandatory): Enter the Kafka topic name.
7. By default, Cora SeSequence uses the parameters configured on the Kafka server. To change the default settings, click **Advanced Options**.
8. If your implementation requires that each message is read and processed before moving on to the next message, select **Acknowledgement**.
9. Click **Finish**.

To learn more about Kafka parameters, refer to the [Kafka documentation](#).