

Apache Kafka Consumer Backends

On This Page:

- [KafkaConsumer.poll and Kafka Stream Entry Points](#)
- [Kafka SimpleConsumer Entry Points](#)

You can configure the AppDynamics Java Agent to detect entry points for Apache Kafka consumer activity using `KafkaConsumer.poll()` (introduced in Kafka v0.10) or Kafka Streams and `KafkaSimpleConsumer.fetch()` (both introduced in Kafka v0.9).

Prior to version 0.11, the Kafka payload did not include a location to store correlation data, so end-to-end Business Transaction correlation is only possible with Kafka client and broker versions 0.11 or greater.

KafkaConsumer.poll and Kafka Stream Entry Points

To instrument Kafka consumer entry points using `KafkaConsumer.poll()` or Kafka Streams, identify the method where the consumer reads messages in a loop in a custom interceptor definition. We instrument the iterator's next method to start and end the BT for each message. There could be many iterators used for iterating messages but we only support iterators that are of type:

- `kafka.consumer.ConsumerIterator`
- `org.apache.kafka.clients.consumer.ConsumerRecords$ConcatenatedIterable$1`

1. Identify the class and method of the loop that processes messages from Kafka. Consider for example a class `MyConsumer` that employs the following loop to poll and process messages from Kafka:

```
private void pollMessages() throws Exception {
    ConsumerRecords<String, String> records = kafkaConsumer.poll(1000);
    for (ConsumerRecord<String, String> record : records) {
        //Processing of the records
        System.out.println(record.value());
    }
}
```

For this case, you want to intercept:

- Class: `MyConsumer`
 - Method: `pollMessages`
2. Use your preferred text editor to create and edit a file named `custom-interceptors.xml` at the following path:
`<agent_home>/<version_number>/conf`

For example:

`/usr/home/appdynamics/appagent/ver4.3.1.0/conf/custom-interceptors.xml`

3. Copy the following XML to `custom-interceptors.xml`:

```
<custom-interceptors>
  <custom-interceptor>
    <interceptor-class-
name>com.singularity.KafkaMarkerMethodInterceptor</interceptor-class-name>
    <match-class type="matches-class">
      <name filter-type="equals">my-fully-qualified-class-name</name>
    </match-class>
    <match-method>
      <name>my-method-name</name>
    </match-method>
  </custom-interceptor>
</custom-interceptors>
```

4. Set the value of the class name to the name of your consumer class. For instance, to specify the `MyConsumer` class:

```
<match-class type="matches-class">
  <name filter-type="equals">com.mycompany.mypackage.MyConsumer</name>
</match-class>
```

5. Set the value of the method name to the name of your message processing loop method. For instance, to specify the `pollMessages` method:

```
<match-method>
  <name>pollMessages</name>
</match-method>
```

After the Java Agent reads the updated configuration, it detects consumer activity and upstream Kafka queue. The application flow map shows the tier receiving data from the Kafka queue. The Kafka queue does not appear on the BT level flow map.

Kafka SimpleConsumer Entry Points

To enable consumer entry points for Kafka clients that retrieve messages using `SimpleConsumer.fetch()`, register the `enable-kafka-consumer` node property with a value of "true".



Kafka consumer activity shows up as an exit call in this case.