



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Integrated Capture Points Guide

Kafka Capture Point - Debugging

12/14/2025

Kafka Capture Point - Debugging

The **ixn-java-aux.jar** file provides the means to debug Kafka Capture Point without Interaction Server, thus providing a simple and rapid sanity check of the Kafka environment.

You can use the Java class

`com.genesyslab.eservices.interactionserver.capturepoints.CheckCApp` to run Kafka Capture Point as a console Java application in the same way it is done in Interaction Server. It can produce/consume messages to/from Kafka. The only command-line argument is the path to the application settings file. For example:

On Windows:

```
java -cp <path to IXN dir>/lib/ixn-java-aux.jar;<path to IXN dir>/groovy_event_logger/lib/KafkaEventLogger/kafka-clients-2.3.0.jar;<path to IXN dir>/groovy_event_logger/lib/KafkaEventLogger/slf4j-api-1.7.26.jar;<path to IXN dir>/transformation/groovy-all-2.4.15.jar; com.genesyslab.eservices.interactionserver.capturepoints.CheckCApp <path to application settings>
```

On Linux:

```
java -cp <path to IXN dir>/lib/ixn-java-aux.jar:<path to IXN dir>/groovy_event_logger/lib/KafkaEventLogger/kafka-clients-2.3.0.jar:<path to IXN dir>/groovy_event_logger/lib/KafkaEventLogger/slf4j-api-1.7.26.jar:<path to IXN dir>/transformation/groovy-all-2.4.15.jar: com.genesyslab.eservices.interactionserver.capturepoints.CheckCApp <path to application settings>
```

The application settings file must be in JSON format. It follows the generic Genesys format: sections are on the first level and options are on the second level. For example:

```
{
  "check-cp-app": {
    "cp-type": "kafka",
    "cp-options-file": "<path to Kafka Capture Point settings file>",
    "received-messages-dir": "<path to a directory with inbound message files>",
    "commit-inbound": "true"
  },
  "notifications-dir": "<path to a directory with notification files>",
}
```

The following options are available:

- `cp-type` - (Mandatory) Must have `kafka` as the value always.
- `received-messages-dir` - A path to a directory where all the messages read from the inbound topic are stored in. Each message is stored in a separate file with an ordered number as a name. The default value is an empty string, which means messages won't be stored.
- `notifications-dir` - A path to a directory where unsolicited notifications are read from the 'to be sent to the notification' topic. Each file is considered to have one notification. All the characters of a file name, up to the last period symbol, are used as a partition key. The default value is an empty string, which means unsolicited notifications won't be sent.
- `commit-inbound` - If the value is set to `false`, Capture Point will never commit an inbound message;

instead, it will keep processing this message and continuously send out notifications, resulting in an infinite loop. The default value is `true`.

- `cp-options-file` - (Mandatory) A path to the Kafka Capture Point settings file. It follows the generic Genesys format: sections are on the first level and options are on the second level. For example:

```
{
  "settings": {
    "inbound-topic-name": "inbound",
    "processed-topic-name": "processed",
    "error-topic-name": "error",
    "notification-topic-name": "notification",
    "copy-original-properties-in-reply": "false",
    "consumer-receive-timeout": "10000",
    "kafka-server": "10.10.19.160:9092,10.10.19.161:9092,10.10.19.162:9092",
  },
  "consumer-options": {
    "max.poll.interval.ms": 1000,
    "max.poll.records": 20,
    "auto.offset.reset": "earliest",
  },
  "producer-options": {
    "retries": 10,
  }
}
```