

KIP-662: Throw Exception when Source Topics of a Streams App are Deleted


- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

Status

Current state: Accepted

Discussion thread: http://mail-archives.apache.org/mod_mbox/kafka-dev/202008.mbox/%3Cee651abd-8582-b899-892a-31405e56d13c%40confluent.io%3E

JIRA:

 Unable to render Jira issues macro, execution error.

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

Today, when source topics of a Streams app are accidentally deleted (either before the Streams app is started or while the app is running), the Streams clients of the app gracefully shut down themselves. More precisely, the consumer group leader notices the missing source topics from the metadata refresh and triggers a rebalance. During the rebalance the leader encodes error `INCOMPLETE_SOURCE_TOPIC_METADATA` into the assignment. Upon getting this error, a stream thread that participates in the rebalance shuts down itself and writes an `ERROR` log message in its log file. When all stream threads of a Streams client are dead, the Streams client is in state `ERROR`. Eventually, all Streams clients transit to state `ERROR`. Such a graceful shutdown does not give users the chance to react programmatically to the deletion of source topics (other than parsing the log files).

In this KIP, we propose to not gracefully shutdown the Streams app in case of the deletion of source topics but to throw an exception instead. The user can then react on this exception in the uncaught exception handler.

Public Interfaces

We propose to add a new exception to allow users to easily distinguish the deletion of source topics from other exception and to specifically react on it in the uncaught exception handler. The new exception is the following:

```
package org.apache.kafka.streams.errors;

public class MissingSourceTopicException extends StreamsException
```

Proposed Changes

We propose that a stream thread should throw a `MissingSourceTopicException` when it receives error `INCOMPLETE_SOURCE_TOPIC_METADATA` encoded in its assignment instead of gracefully shutting down itself. This exception will not be caught anywhere and will eventually reach the uncaught exception handler, if one is set. When all stream threads of a Streams client are dead, the Streams clients will be in state `ERROR`. The transition to state `ERROR` after the death of all stream threads corresponds to the current behavior and will not be changed by this KIP.

Compatibility, Deprecation, and Migration Plan

The proposed changes do not need any deprecation or migration plan.

The only difference to the current behavior is that the uncaught exception handler is called. If the uncaught exception handler is not set or it is set but does not handle `StreamsExceptions` or its ancestors, the behavior does not change. If the uncaught exception handler is set and handles `StreamsExceptions` or one of its ancestors, that code will be executed when a source topic is deleted which it is currently not the case. However, users do not need to change anything to run their Streams app. They will merely get an additional error notification for a case that actually is an error.

Rejected Alternatives

none