

KIP-260: add primary join operation for Stream-Stream join (WIP)

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

Status

Current state: *"Under Discussion"*

Discussion thread: [here](#) [Change the link from the KIP proposal email archive to your own email thread]

JIRA: [here](#) [Change the link from KAFKA-1 to your own ticket]

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

Motivation

Currently in the Stream-Stream window joining context, the joiner will take in two incoming data streams, for example A and B. Upon new record arrival from A/B, joiner will apply exhaustive search on window store B/A to get the all the possible joining records. However, the joiner could not optimize the case where we only care about exactly one match, or say the first seen. We are proposing a new API which could enforce a joining called "FirstOrderJoin" which only returns the first seen key-value pair on the other side.

Public Interfaces

We need to make change to RocksdbStore.java.

Briefly list any new interfaces that will be introduced as part of this proposal or any existing interfaces that will be removed or changed. The purpose of this section is to concisely call out the public contract that will come along with this feature.

A public interface is any change to the following:

- Binary log format
- The network protocol and api behavior
- Any class in the public packages under clientsConfiguration, especially client configuration
 - org/apache/kafka/common/serialization
 - org/apache/kafka/common
 - org/apache/kafka/common/errors
 - org/apache/kafka/clients/producer
 - org/apache/kafka/clients/consumer (eventually, once stable)
- Monitoring
- Command line tools and arguments
- Anything else that will likely break existing users in some way when they upgrade

Proposed Changes

Describe the new thing you want to do in appropriate detail. This may be fairly extensive and have large subsections of its own. Or it may be a few sentences. Use judgement based on the scope of the change.

Compatibility, Deprecation, and Migration Plan

- What impact (if any) will there be on existing users?
- If we are changing behavior how will we phase out the older behavior?
- If we need special migration tools, describe them here.
- When will we remove the existing behavior?

Rejected Alternatives

If there are alternative ways of accomplishing the same thing, what were they? The purpose of this section is to motivate why the design is the way it is and not some other way.