

KIP-406: GlobalStreamThread should honor custom reset policy

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

Status

Current state: *Under Discussion*

Discussion thread: [here](#)

JIRA: [here](#)

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

Motivation

Currently, when GlobalStreamThread receives an InvalidOffsetException from global topics, it will be forwarded to the user which is used to indicate that something wrong had occurred and user needs to restart the application for that instance. We have improved upon the durability of GlobalStreamThread previously by wiping all data that had already been processed and rebuilding the store entirely (which was done in [KAFKA-6121](#)). However, we need to handle this exception automatically(as we do we in Stream Threads) so that the user does not need to restart the application whenever such a scenario arises. So, in nutshell the main motivation for this KIP is to let the runtime auto-recover instead of dying as it does currently.

Public Interfaces

- A new KIP will be created to change the state machine so that global state stores can also be restored in a RESTORING state at the stream client level, defined as "at least one of the state stores assigned to this client, either global or non-global, is still restoring", and emphasize that during this state the client may still be able to process records, just probably not in full-speed.

Proposed Changes

- Create a new independent KIP to change the Kafka Stream's state machine.
- Cleanup stores in GlobalStreamThread and change the state to restoring to rebuild the stores from the "earliest" offset.
- Add javadocs to ensure that the global topics are compacted in order to ensure correct semantics.

Compatibility, Deprecation, and Migration Plan

- The compatibilities for the state machine changes will be handled by the new KIP.
- Since the InvalidOffsetException will not throw StreamsException from InvalidOffsetException anymore, no specific changes required from users. Although StreamsException will not be thrown from this flow but there are other places from which StreamsException is still thrown.

Rejected Alternatives

- *Wipe the store and seekToBeginning and log the error in case of InvalidOffsetException.*
- *Provide a reset policy to reset offsets to earliest/latest as resetting to latest doesn't really make sense for global stores.*
- *Change the state machine to include "GlobalRestoring" state so that StreamThreads are paused from processing till the time global stores are completely restored.*
- *Per-topic reset policy is not required, since we not provided the option to reset to various policies to users.*