

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 dies from a fatal error, an InvalidOffsetException will be forwarded to the user which is used to indicate that something wrong had occurred. 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 should also give the user the option of choosing their reset policy which up to this time, they had no chance of selecting.

Public Interfaces

A new reset policy(s) will be added which will allow the user some flexibility in determining the resilience of GlobalStreamThread. It should also be noted that `Topology.addGlobalStore` does not allow the user to decide which policy to use. This method will potentially have to be extended.

Proposed Changes

TBD. The details of the new reset policy is yet to be decided upon.

Compatibility, Deprecation, and Migration Plan

- The current `Topology.addGlobalStore` method will most likely be extended. One option will be adding a second version of `addGlobalStore` with extra input parameters while the original method calls the added method with default values. This would not force the user to change their preexisting code in any way.

Rejected Alternatives

N/A