

KIP-654: Aborted transaction with non-flushed data should throw a non-fatal exception

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

Status

Current state: *Accepted*

Discussion thread: <https://lists.apache.org/thread.html/r2ef8159a094c0d1b01b217174069342483afb5c71d37fd32e8bc4c81%40%3Cdev.kafka.apache.org%3E>

JIRA: [KAFKA-10186](#) [*Aborting transaction with pending data should throw non-fatal exception*]

Motivation

Currently in the Java client producer, if you try to abort a transaction with any pending (non-flushed) data, the exception thrown is a

```
KafkaException("Failing batch since transaction was aborted")
```

This exception type is generally considered fatal.

However, this is a valid state to be in. The point of throwing the exception is to alert that the records will not be sent, not that you are in an unrecoverable error state.

We should potentially throw a different type of exception here to alert the user to the aborted transaction. This will also distinguish this from fatal and recoverable errors.

Public Interfaces

We propose a new Exception to be thrown for an aborted transaction. This will help distinguish this particular error from the others and help the clients retry the failed batch instead of handling this as an irrecoverable exception.

This new exception will be thrown in addition to the `KafkaException` exceptions that are thrown when the transaction is unrecoverable.

Proposed Changes

We propose a `TransactionAbortedException` that will be thrown in case the transaction is aborted but still recoverable. This will allow users to check this exception and retry if needed.

As this will extend `ApiException`, users already catching `KafkaException` (which `ApiException` extends) will not be impacted and hence this is a backward compatible change.

KafkaProduceTransactionException

```
public class TransactionAbortedException extends ApiException {

    private final static long serialVersionUID = 1L;

    public TransactionAbortedException(String message, Throwable cause) {
        super(message, cause);
    }

    public TransactionAbortedException(String message) {
        super(message);
    }

    public TransactionAbortedException() {
        super("Failing batch since transaction was aborted");
    }
}
```

Compatibility, Deprecation, and Migration Plan

- *What impact (if any) will there be on existing users?*
 - Existing users will see a new kind of exception that is still caught by their code since this will extend `ApiException` and by extension `KafkaException`
- *If we need special migration tools, describe them here.*
 - None needed.
- *When will we remove the existing behavior?*
 - This could be removed in the release after this is adopted and mentioned clearly in the release notes.

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.