

KIP-102 - Add close with timeout for consumers

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Test Plan](#)
- [Rejected Alternatives](#)
 - [Use a small hard-coded timeout](#)
 - [Use a Long.MAX_VALUE as default timeout](#)

Status

Current state: *Accepted*

Discussion thread: [here](#)

JIRA: [KAFKA-4426](#)

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

Motivation

Consumer API currently has one `close()` method without a timeout. [KAFKA-3703](#) added graceful close of the consumer with a hard-coded timeout of 5 seconds to ensure that consumers complete pending commits and leave the group gracefully before closing connection to the coordinator.

Producer interface has two `close()` methods.

Produce close methods

```
public void close();
public void close(long timeout, TimeUnit unit);
```

The first variant waits for graceful close without a time limit (`timeout=Long.MAX_VALUE`). The second attempts to close gracefully, but terminates the produce, aborting any pending requests after the timeout.

For consistency with producers, consumers should also expose the same two variants for close.

Public Interfaces

A new close method with timeout will be added to the `Consumer` interface. The existing `close()` method will attempt to close gracefully with a default timeout of 30 seconds.

Consumer close with timeout

```
/**
 * Tries to close the consumer cleanly within the specified timeout. If the close does not complete within the
 * timeout, force close the consumer.
 */
public void close(long timeout, TimeUnit unit);
```

Proposed Changes

Both `close` methods will attempt to commit offsets if necessary and leave group before the coordinator connection is closed. The maximum time allowed to complete each request initiated during close will be the request timeout. This ensures that close with large timeout doesn't wait forever if requests are never completed, for example when brokers are shutdown. Note that this KIP only adds timeouts to the close code path. Unbounded waits in the consumer during normal operation are not addressed by this KIP.

The new `close(long timeout, TimeUnit unit)` method will use the specified timeout as the maximum time for attempting to close gracefully. If commit or leave group request is not complete within the timeout, the connection is closed and the requests may not be sent to the broker.

The existing `close()` method without a timeout will attempt to close the consumer gracefully with a default timeout of 30 seconds. This is different from the producer default of `Long.MAX_VALUE` since consumers don't have large requests. Applications can specify a timeout of `Long.MAX_VALUE` to attempt graceful close without a timeout. Since commit and leave group requests are timed out after the request timeout, the upper bound will be approximately $2 * \text{request.timeout.ms}$ (around 10 minutes by default).

The hard-coded 5 second timeout to send requests to the broker before closing the coordinator connection will be removed. This will be replaced with the lower of request timeout or the time remaining from the close timeout specified.

Compatibility, Deprecation, and Migration Plan

What impact (if any) will there be on existing users?

Versions 0.10.1.0 and earlier closed consumers without waiting for commits or leave group requests to be sent. With this change, `consumer.close()` method waits for all pending requests to be sent for upto 30 seconds and hence `close()` could take longer to complete.

Test Plan

Unit tests added in [KAFKA-3703](#) to test the hard-coded timeout will be altered to work with the new close timeout. Existing integration and system tests already test close with default timeout.

Rejected Alternatives

Use a small hard-coded timeout

A single close method with a small hard-coded timeout can be used to attempt to close gracefully if possible. But this is inconsistent with the `Producer` where applications can choose to wait for a longer time or not wait at all for graceful close.

Use a `Long.MAX_VALUE` as default timeout

`Producer.close()` uses `Long.MAX_VALUE` as default timeout. Consumers could also use the same default. But this could result in waits of upto 10 minutes for the default `close()` method to complete with the default request timeout of 5 minutes. Since consumers only attempt to commit offsets and leave group during `close()` on a best-effort basis, a smaller default value is more suitable.