

Kafka Exactly Once - Dealing with older message formats when idempotence is enabled

Background

In the discussion of [KIP-185: Make exactly once in order delivery per partition the default producer setting](#), it was realized that we don't have graceful handling when an idempotence-enabled producer is writing to a broker with a message format older than v2 (ie. the 0.11.0 message format).

In particular, if we enable idempotence, any produce requests to topics with an older message format will fail with an `UnsupportedVersionException`. Thus if the idempotent producer was to be made the default, the out of the box producer would fail to produce when used with clusters which haven't upgraded the message format yet.

This is particularly problematic since the recommended upgrade path is to upgrade broker code while keeping the message format at the older version, then upgrade all clients, and only finally upgrade the message format on the server. With the current behavior, the middle step is actually untenable if we enable idempotence as the default.

Design

To solve this problem, we propose introducing a new idempotence mode in 1.0.0, ie. `enable.idempotence=requested`. The other modes would be `enable.idempotence=off` and `enable.idempotence=required`. In the `requested` mode, idempotence would be best effort, ie. you get it if the broker supports it, otherwise no error is raised. In `required` mode, requests would fail if the broker did not support idempotence. In `off` mode, there is no idempotence (similar to the `false` value today).

The new `enable.idempotence=requested` mode would be introduced in the 1.0.0 release. In addition, the 1.0.0 client will disable idempotence if the message format of the destination topic doesn't support it.. In `required` mode, the client will raise an error if it detects an incompatible message format.

This means we would need to upgrade the `TopicMetadata` in the `MetadataResponse` version to include the version of the message format for the topic in question.

Additionally, if a 1.0.0 client with `enable.idempotence=requested` is writing to an 0.11.0 broker, idempotence will be disabled.

Finally, a 1.0.0 producer would interpret `enable.idempotence=true` as `enable.idempotence=required` and `enable.idempotence=false` as `enable.idempotence=off`.

Level of Effort

1. Update `MetadataResponse` to include the message format version - 1 day
2. Use the message format version to determine whether a partition supports idempotence - 5 days.
3. Client side changes to send the idempotent producer mode, interpret old configurations of `enable.idempotence`— 0.5 days.

Total: 6.5