KIP-948: Allow custom prefix for internal topic names in Kafka Streams

This page is meant as a template for writing a KIP. To create a KIP choose Tools->Copy on this page and modify with your content and replace the heading with the next KIP number and a description of your issue. Replace anything in italics with your own description.

Status

Current state: Under Discussion

Discussion thread: *here*

JIRA: None yet

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

Motivation

Currently, the application.id config is used as the consumer group id for the underlying KafkaConsumer and as a prefix for internal topics. In most cases, this is not an issue, but some use cases require more flexibility, i.e. different naming conventions (prefixes) for topics and consumer groups might be needed.
Real-world example: In a (highly automated) multitenant setup, each team using a central Kafka cluster has a dedicated topic namespace and can control ACLs for the topics in that namespace. To be able to manage ACLs for internal topics of their KafkaStreams app, application.id config must match the team's topic namespace prefix, and this is not always desirable.
It would be nice if the internal topic prefix could be additionally configured.

Public Interfaces

- New config “internal.topics.prefix” introduced for Kafka Streams
- The new argument “internal.topics.prefix” added to the Kafka Streams reset tool

Proposed Changes

Add new KafkaStreams config - internal.topics.prefix, which would be used for internal topics generation. This prefix would be added at the beginning of the internal topic name generated by the current algorithm. The default value for this new config would be an empty string.
This means that application.id is still used as a part of the internal topic names, but an additional prefix can be added. That way, backward compatibility is preserved.

Considerations:

- Kafka Streams application reset tool would be affected - "internal.topics.prefix" config should also be added to its arguments and must be passed along the application.id argument in case Kafka Streams application had custom value for internal.topics.prefix.
- Changing the "internal.topics.prefix" config value of the existing Kafka Streams app without changing its application.id would introduce inconsistency since internal topics would be regenerated, but the consumer group id would stay the same, meaning input topics would not be reprocessed from the beginning. This should be emphasized in the doc.

Compatibility, Deprecation, and Migration Plan

- This change is backward compatible since, with the default value (empty string), it doesn't change the internal topics name generation algorithm.

Test Plan

Describe in few sentences how the KIP will be tested. We are mostly interested in system tests (since unit-tests are specific to implementation details). How will we know that the implementation works as expected? How will we know nothing broke?

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.