

KIP-594: Expose output topic names from TopologyTestDriver


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

Status

Current state: Accepted

Discussion thread: [here](#)

JIRA:

 Unable to render Jira issues macro, execution error.

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

Motivation

When using the TopologyTestDriver to assert topologies are working as expected, it can be useful to know the full set of topics the topology produces to during a test run - including internal changelog and repartition topics. This enables valuable catch-all assertions to make sure the application doesn't produce to *unexpected* topics in addition to the regular assertions that the application does produce the *expected* data.

Public Interfaces

Addition of a new method to TopologyTestDriver called producedTopicNames, that will return the name of all topics the topology produced messages to during the test run, i.e. the set of topics the existing createOutputTopic method can be called with, that will return a topic with data available. This set includes internal and sink topics.

```
/**
 * Get the names of all the topics to which records have been produced.
 * <p>
 * Call this method after piping the input into the test driver to retrieve the full set of topics the topology
 * produced records to during the test run.
 * <p>
 * The returned set of topic names includes changelog, repartition and sink topic names.
 *
 * @return the set of topic names the topology has produced to.
 */
public final Set<String> producedTopicNames(){...}
```

Proposed Changes

```
public final Set<String> producedTopicNames() {
    return Collections.unmodifiableSet(outputRecordsByTopic.keySet());
}
```

Compatibility, Deprecation, and Migration Plan

None

Rejected Alternatives

Pre-populate the set of topic names returned by `produceTopicNames` by inspecting the `TopologyDescription` on construction of the `TopologyTestDriver` instance.

This was rejected as it would not handle the case of dynamic topic bindings, i.e. non-static `TopicNameExtractors`. Where these are used, the topic name is not known until run time. Hence a purely `TopologyDescription`` driven implementation would completely fulfil the use case.

A future KIP may still cover extending the `TopologyDescription`` with convenience methods to get the set of input, internal and output topic names, as these was considered complementary to this KLIP.