


# KIP-234: add support for getting topic defaults from AdminClient

- [Status](#)
- [Motivation](#)
  - [Current State/Workarounds](#)
    - [Create then Edit](#)
    - [Know What Broker Configs Are](#)
      - [Always Configure ALL Configs For a Topic](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

## Status

**Current state:** *Under Discussion*

**Discussion thread:** [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 creating a topic a user may want to know what the broker defaults will be for their topic before creating it.

## Current State/Workarounds

### Create then Edit


A user could create the topic as they see fit. Once it is created they could then inspect the topic to determine it's configs. And then they would be able to alter it to suit their needs. This is problematic because there is a period where your topic is in the incorrect (and possibly invalid) state. E.g., we want a `compact` topic, but the broker default is `delete`. We create a topic with defaults, then read the configs, then update the configs to what we want. In the intervening time a user could have produced a message that is invalid for the topic, i.e. a message without a key. Similar things could happen with a number of settings, `retention.ms`, [`message.timestamp.difference.max.ms`](#), `message.timestamp.type`, etc.

### Know What Broker Configs Are

A user could know, or inspect from the broker, the current broker settings. They would then need to know and map those extant configs to the proper topic configs. There is no chance for inconsistent topic creation, but we are now putting the onus on the user of AdminClient to understand the internal workings of the KafkaConfig. E.g., the mappings from broker config to topic config ([defined here](#)), cascading elements like `log.retention.{hours,minutes,ms}`, etc.

### Always Configure ALL Configs For a Topic

A user would need to supply **all** configs for a topic. This requires a user to 1) know all configs that they might need to set, 2) lose the ability to purposefully leave a config unset during topic creation to retain the ability to use broker default updates. E.g., A user may want to create a `compact` topic but not have the experience/knowledge to know what a good value for `min.cleanable.dirty.ratio` would be, so they decide to leave it as default. The admin who is actually running the cluster gets paged and after some testing realizes they can update the broker defaults (hopefully using [KIP-226 - Dynamic Broker](#)

[Configuration](#)) and fix all affected topics (similar to  Unable to render Jira issues macro, execution error. ), without going through and changing all the topic configs on their cluster 1 by 1.

# Proposed Changes

We propose allowing ``name=null`` for a ``ConfigResource`` with ``type=TOPIC``. This will allow us to get the default topic configs from the broker through an already existing path.

[WIP commit with changes.](#)

Sample usage/output:

```

// i set the broker default cleanup.policy=compact,delete

ConfigResource topicResource = new ConfigResource(Type.TOPIC, null);
AdminClient kafkaAdminClient = new KafkaAdminClient(...);
Map<ConfigResource, Config> configs = kafkaAdminClient.describeConfigs(Collections.singleton(topicResource)).
all().get(15, TimeUnit.MILLISECONDS);
System.out.println(configs.toString());
...
{
    ConfigResource{type=TOPIC, name='null'}=
        Config(entries=[
            ConfigEntry(name=flush.messages, value=9223372036854775807, isDefault=true,
isSensitive=false, isReadOnly=false),
            ConfigEntry(name=message.timestamp.type, value=CreateTime, isDefault=true,
isSensitive=false, isReadOnly=false),
            ConfigEntry(name=preallocate, value=false, isDefault=true, isSensitive=false,
isReadOnly=false),

            ConfigEntry(name=cleanup.policy, value=compact,delete, isDefault=true,
isSensitive=false, isReadOnly=false),

            ConfigEntry(name=segment.bytes, value=1073741824, isDefault=true, isSensitive=false,
isReadOnly=false),
            ConfigEntry(name=delete.retention.ms, value=86400000, isDefault=true,
isSensitive=false, isReadOnly=false),
            ConfigEntry(name=segment.ms, value=604800000, isDefault=true, isSensitive=false,
isReadOnly=false),
            ConfigEntry(name=min.insync.replicas, value=1, isDefault=true, isSensitive=false,
isReadOnly=false),
            ConfigEntry(name=file.delete.delay.ms, value=60000, isDefault=true, isSensitive=false,
isReadOnly=false),
            ConfigEntry(name=retention.ms, value=604800000, isDefault=true, isSensitive=false,
isReadOnly=false),
            ConfigEntry(name=max.message.bytes, value=1000012, isDefault=true, isSensitive=false,
isReadOnly=false),
            ConfigEntry(name=message.format.version, value=1.0-IV0, isDefault=true,
isSensitive=false, isReadOnly=false),
            ConfigEntry(name=index.interval.bytes, value=4096, isDefault=true, isSensitive=false,
isReadOnly=false),
            ConfigEntry(name=retention.bytes, value=-1, isDefault=true, isSensitive=false,
isReadOnly=false),
            ConfigEntry(name=segment.index.bytes, value=10485760, isDefault=true,
isSensitive=false, isReadOnly=false),
            ConfigEntry(name=segment.jitter.ms, value=0, isDefault=true, isSensitive=false,
isReadOnly=false),
            ConfigEntry(name=compression.type, value=producer, isDefault=true, isSensitive=false,
isReadOnly=false),
            ConfigEntry(name=min.cleanable.dirty.ratio, value=0.5, isDefault=true,
isSensitive=false, isReadOnly=false),
            ConfigEntry(name=min.compaction.lag.ms, value=0, isDefault=true, isSensitive=false,
isReadOnly=false),
            ConfigEntry(name=unclean.leader.election.enable, value=false, isDefault=true,
isSensitive=false, isReadOnly=false),
            ConfigEntry(name=message.timestamp.difference.max.ms, value=9223372036854775807,
isDefault=true, isSensitive=false, isReadOnly=false),
            ConfigEntry(name=flush.ms, value=9223372036854775807, isDefault=true,
isSensitive=false, isReadOnly=false),
            ConfigEntry(name=follower.replication.throttled.replicas, value=, isDefault=true,
isSensitive=false, isReadOnly=false),
            ConfigEntry(name=leader.replication.throttled.replicas, value=, isDefault=true,
isSensitive=false, isReadOnly=false)
        ])
}

```

# Compatibility, Deprecation, and Migration Plan

## Rejected Alternatives

- **New AdminClient Method.** This would expand the surface area of AdminClient and the Protocol with benefit of being more explicit.
- **Return config information from AdminClient.createTopics(validate=true).** This would piggyback on a common user pattern of (come up with possible configs, validate, repeat), but would require much larger changes to the protocol since `createTopics` returns void now.