

# KIP-840: Config file option for MessageReader/MessageFormatter in ConsoleProducer/ConsoleConsumer

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
  - [Simple example](#)
  - [Mixed example](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Test Plan](#)
- [Rejected Alternatives](#)

## Status

**Current state:** ACCEPTED (4x+1 binding, 1x+1 non-binding)

**Discussion thread:** [here](#)

**Vote thread:** [here](#)

**JIRA:**  Unable to render Jira issues macro, execution error.

**Pull Request:** [here](#)

## Motivation

Tools `kafka-console-producer.sh` and `kafka-console-consumer.sh` have the multiple option `--property` to setup the `MessageReader` for Producer (defined by `--line-reader`) or the `MessageFormatter` for Consumer (defined by `--formatter`).

To setup the `KafkaProducer/KafkaConsumer` they have the multiple option `--producer-property/--consumer-property` **and** the config file option `--consumer.config/--producer.config`.

The idea is to offer the same possibility to use a config file option to setup the `MessageReader/MessageFormatter` in addition to the existing option `--property`.

It will be useful in different use cases:

- when having multiple `--property` (the default reader can have up to 11 properties)
- when passing properties with sensitive data (like `schema.registry.ssl.truststore.password` for `kafka-avro-console-producer`) (could also be mitigated by using env vars)

## Public Interfaces

Add an option:

- `--reader-config` to tool `kafka-console-producer.sh`
- `--formatter-config` to tool `kafka-console-consumer.sh`

As for `--producer-property/--consumer-property` with `--consumer.config/--producer.config`, any value from option `--property` would override value from config file.

## Simple example

- Before (multiple `--property` options):

```
kafka-console-producer.sh [...] --property print.timestamp=true \
--property print.key=true \
--property print.offset=true \
--property print.partition=true \
--property print.headers=true \
--property print.value=true \
--property key.separator=':' \
--property line.separator=';' \
--property headers.separator='|'
```

- After (all properties in one file):

#### reader-config.properties

```
print.timestamp=true
print.key=true
print.offset=true
print.partition=true
print.headers=true
print.value=true
key.separator=:
line.separator=;
headers.separator=|
```

```
kafka-console-producer.sh [...] --reader-config reader-config.properties
```

## Mixed example

When a property is in the config file **and** as option, option wins.

#### reader-config.properties

```
print.timestamp=true
print.key=true
print.offset=true
print.partition=true
print.headers=true
print.value=true
key.separator=:
line.separator=;
headers.separator=|
```

```
kafka-console-producer.sh [...] --reader-config reader-config.properties --property key.separator='/'
```

Then, the MessageReader will get `key.separator=/'`, not `key.separator=:`

## Proposed Changes

Implement the new options as described in [Public Interfaces](#) section:

- ConsoleProducer: new option "reader-config" with description "Config properties file for the message reader. Note that --property takes precedence over this config."
- ConsoleConsumer: new option "formatter-config" with description "Config properties file to initialize the message formatter. Note that --property takes precedence over this config."

Already implemented in [pull request #12175](#).

## Compatibility, Deprecation, and Migration Plan

No impact on existing users, just a new possibility available.

## Test Plan

Launch the tools with the new config file option and check that properties from config file are applied to the reader/formatter.

Define a property in config file and in options and check that option override config file.

## Rejected Alternatives

- Use of generic option `--config` to avoid confusion in the purpose of this config file (but on the other hand, the current `--property` is quite generic already, maybe another KIP to change it).
- Use of option name `--read.config/--formatter.config`, with "." as separator, as it's not consistent with the commonly used option `--command-config`