

KIP-363: Allow performance tools to print final results to output file

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)
 - [Behavior:](#)
 - [Example:](#)
- [Compatibility, Deprecation, And Migration Plan](#)
- [Test Plan](#)
- [Rejected alternatives](#)
 - [Make delimiters configurable](#)
- [Future Work](#)

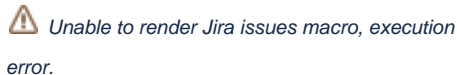
This page is meant as a template for writing a [KIP](#). To create a KIP choose Tools->Copy on this page and modify with your content and replace the heading with the next KIP number and a description of your issue. Replace anything in italics with your own description.

Status

Current state: *Under Discussion*

Discussion thread: [here](#)

JIRA:

A screenshot of a JIRA error message. It features a yellow warning triangle icon on the left, followed by the text "Unable to render Jira issues macro, execution error." in a dark blue font.

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

Motivation

Currently, ProducerPerformance and ConsumerPerformance tools do not provide command line options to save final results into an output file.

It would be useful to allow users to generate performance reports in a machine-readable format, and save final results to a file. This way, results could be easily processed by external applications. To keep this KIP short and easy to review, only CSV format will be supported; other formats (such as JSON, XML, etc.) can be introduced by follow-up KIP(s).

[RFC-4180](#) defines a formal specification of the CSV format. It allows an optional header line that contain names corresponding to the fields. In certain situations, a header line might be beneficial to Kafka users too.

Public Interfaces

Add optional arguments "output-path" and "output-with-header" to both ProducerPerformance and ConsumerPerformance with the following specification

```
--output-with-header Print out final results to output file with header. (default: false)
--output-path OUTPUT-PATH Write final results to the file OUTPUT-PATH.
```

Proposed Changes

In ProducerPerformance:

```

parser.addArgument("--output-path")
    .action(store())
    .required(false)
    .type(String.class)
    .metavar("OUTPUT-PATH")
    .dest("outputPath")
    .help("Write final results (excluding metrics) to the file specified by OUTPUT-PATH.");

parser.addArgument("--output-with-header")
    .action(storeTrue())
    .required(false)
    .type(Boolean.class)
    .dest("outputWithHeader")
    .help("Print out final results to output file with headers.");

```

In ConsumerPerformance:

```

val outputWithHeaderOpt = parser.accepts("output-with-header", "Print out final results to output file with headers.")
val outputPathOpt = parser.accepts("output-path", "Write final results (excluding metrics) to the specified file.")
    .withOptionalArg()
    .describedAs("output file")
    .ofType(classOf[String])

```

Behavior:

- When "--output-path" is specified by user, final results of ProducerPerformance and ConsumerPerformance will be printed not only to the standard output but also to the given file in **CSV** format. An exception will be thrown if the file already exists.
- When "--output-with-header" is specified, a header record will be also printed into the output file (it will be the first line of the output). This argument only takes effect if "--output-path" is also specified.

Example:

Running ProducerPerformance tool with the following options:

```

bin/kafka-run-class.sh org.apache.kafka.tools.ProducerPerformance --topic test --num-records 1000000 --throughput -1 --record-size 100 --producer-props bootstrap.servers=localhost:9092 --output-path producer_stats.csv --output-with-header

```

will generate an output file called producer_stats.csv in CSV format:

```

records sent,records/sec,MB/sec,ms avg latency,ms max latency,ms 50th,ms 95th,ms 99th,ms 99.9th
1000000,263713.0801687764,25.14963914573444,430.092296,873.0,490,801,870,873

```

without --output-with-header, only final results are printed

```

1000000,263713.0801687764,25.14963914573444,430.092296,873.0,490,801,870,873

```

Compatibility, Deprecation, And Migration Plan

There won't be any change of current behavior. New arguments of ProducerPerformance and ConsumerPerformance are optional.

Test Plan

Review existing unit tests and implement new test cases that cover new functionality.

Rejected alternatives

Make delimiters configurable

To limit the scope, this KIP only targets CSV output format. [RFC-4180](#) describes the format, and it does not allow the configuration of delimiters.

Future Work

Allowing users to specify other output formats (such as JSON or XML) might be beneficial in some use cases.

Overriding output file might be also useful occasionally.