

# KIP-756: Move StreamsResetter tool outside of core

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

## Status

**Current state:** *Discarded* (covered by [KIP-906: Tools migration guidelines](#))

**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

StreamsResetter tool is currently located at `core` module because of historic reasons, mainly initial dependency with Zookeeper client.

This tool has been requested by users to call it from Streams application, therefore this KIP is aiming to:

- Create a new API for StreamsResetter.
- Move the CLI from core to tools.

## Proposed Changes

New `StreamsResetter` API:

```
public class StreamsResetter {
    public static class StreamsResetterOptions {

        final ResetOffsetsOptions resetOffsetsOptions;

        boolean forceConsumerRemoval = false;
        boolean dryRun = false;

        public StreamsResetterOptions(final ResetOffsetsOptions resetOffsetsOptions) {
            this.resetOffsetsOptions = resetOffsetsOptions;
        }

        public static StreamsResetterOptions with(final ResetOffsetsOptions resetOffsetsOptions) {
            return new StreamsResetterOptions(resetOffsetsOptions);
        }

        public static StreamsResetterOptions with(final ResetOffsetsOptions resetOffsetsOptions,
                                                  final boolean forceConsumerRemoval,
                                                  final boolean dryRun) {
            final StreamsResetterOptions opts = new StreamsResetterOptions(resetOffsetsOptions);
            opts.forceConsumerRemoval = forceConsumerRemoval;
            opts.dryRun = dryRun;
            return opts;
        }
    }

    public static class ResetOffsetsOptions {
```

```

public static final String DATETIME_FORMAT = "yyyy-MM-dd'T'HH:mm:ss.SSS";

Optional<Long> toOffset = Optional.empty();
Optional<String> toDatetime = Optional.empty();
Optional<String> byDuration = Optional.empty();
boolean toEarliest = false;
boolean toLatest = false;
Optional<String> fromFile = Optional.empty();
Optional<Long> shiftBy = Optional.empty();

private ResetOffsetsOptions() {
}

public static ResetOffsetsOptions withDefaults() {
    final ResetOffsetsOptions resetOffsetsOptions = new ResetOffsetsOptions();
    resetOffsetsOptions.toEarliest = true;
    return resetOffsetsOptions;
}

public static ResetOffsetsOptions toEarliest() {
    final ResetOffsetsOptions resetOffsetsOptions = new ResetOffsetsOptions();
    resetOffsetsOptions.toEarliest = true;
    return resetOffsetsOptions;
}

public static ResetOffsetsOptions toLatest() {
    final ResetOffsetsOptions resetOffsetsOptions = new ResetOffsetsOptions();
    resetOffsetsOptions.toLatest = true;
    return resetOffsetsOptions;
}

public static ResetOffsetsOptions fromFile(final String path) {
    final ResetOffsetsOptions resetOffsetsOptions = new ResetOffsetsOptions();
    resetOffsetsOptions.fromFile = Optional.of(path);
    return resetOffsetsOptions;
}

public static ResetOffsetsOptions toDatetime(final String datetime) {
    final ResetOffsetsOptions resetOffsetsOptions = new ResetOffsetsOptions();
    final SimpleDateFormat datetimeFormat = new SimpleDateFormat(DATETIME_FORMAT);
    resetOffsetsOptions.toDatetime = Optional.of(datetimeFormat.format(datetime));
    return resetOffsetsOptions;
}

public static ResetOffsetsOptions byDuration(final String duration) {
    final ResetOffsetsOptions resetOffsetsOptions = new ResetOffsetsOptions();
    resetOffsetsOptions.byDuration = Optional.of(duration);
    return resetOffsetsOptions;
}

public static ResetOffsetsOptions toOffset(final Long offset) {
    final ResetOffsetsOptions resetOffsetsOptions = new ResetOffsetsOptions();
    resetOffsetsOptions.toOffset = Optional.of(offset);
    return resetOffsetsOptions;
}

public static ResetOffsetsOptions shiftBy(Long offsets) {
    final ResetOffsetsOptions resetOffsetsOptions = new ResetOffsetsOptions();
    resetOffsetsOptions.shiftBy = Optional.of(offsets);
    return resetOffsetsOptions;
}
}

public StreamsResetter(final Properties properties,
    final String groupId,
    final List<String> inputTopics) {
    this(properties, groupId, inputTopics, Collections.emptyList(), Collections.emptyList());
}

public StreamsResetter(final Properties properties,
    final String groupId,

```

```

        final List<String> inputTopics,
        final List<String> intermediateTopics) {
    this(properties, groupId, inputTopics, intermediateTopics, Collections.emptyList());
}

public StreamsResetter(final Properties properties,
    final String groupId,
    final List<String> inputTopics,
    final List<String> intermediateTopics,
    final List<String> internalTopics) {
    this.properties = properties;
    this.groupId = groupId;
    this.inputTopics = inputTopics;
    this.intermediateTopics = intermediateTopics;
    this.internalTopics = internalTopics;
}

public int resetStreams(final StreamsResetterOptions options) {
    return resetStreams(options, new Properties());
}

public int resetStreams(final StreamsResetterOptions options, final Properties consumerConfigs) {
    // Implementation
}
}

```

#### Usage:

```

// Defaultnew StreamsResetter(properties, appID, List.of("input"), StreamsResetterOptions.with
(ResetOffsetOptions.withDefaults());

```

Turn current StreamsResetter into StreamsResetterCommand and move it into `tools` module. This will require to add a dependency between tools and streams modules.

## Compatibility, Deprecation, and Migration Plan

At the moment, StreamsResetter has been marked as an "unstable" API, which suppose non backward compatibility guarantees.

## 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.*