

KIP-171 - Extend Consumer Group Reset Offset for Stream Application

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

Status

Current state: *Adopted*

Discussion thread: [here](#)

JIRA: [here](#)

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

Motivation

This KIP is aimed to extend current Consumer Group Reset Offset tooling (implemented by KIP-122[1]) for Kafka Streams applications.

Currently ``kafka-streams-application-reset`` only supports returning to the earliest offset on input topics. But ``kafka-consumer-groups`` ``reset-offsets`` tool support more options detailed in KIP-122.

This KIP is considering include current ``reset-offsets`` scenarios on ``kafka-streams-application-reset`` to have more options over input-topics offsets.

Public Interfaces

"kafka-streams-application-reset" supports the current features[2]:

1. for any specified input topic, it resets all offsets to zero
2. for any specified intermediate topic, seeks to the end for all partitions
3. for all internal topic
 - 3.1. resets all offsets to zero
 - 3.2. deletes the topic

This KIP is considering adding these options:

Scenarios

Only one scenario should be defined for ``input-topics`` to proceed with the execution. If not, ``to-earliest`` will be used by default.

1.	Reset to Datetime	<code>--to-datetime</code> YYYY-MM-DDTHH:mm:SS.sss±hh:mm <code>--to-datetime</code> YYYY-MM-DDTHH:mm:SS.sssZ <code>--to-datetime</code> YYYY-MM-DDTHH:mm:SS.sss	Datetime must be specified in ISO8601 format. This option will translate the datetime to Epoch milliseconds, find the offsets by timestamp, and reset to those offsets. If the Timezone is not specified, it will use UTC.	Reset to first offset since 01 January 2017, 00:00:00 hrs <code>--application-id app1 --input-topics foo --to-datetime 2017-01-01T00:00:00.000Z</code>
2.	Reset by Duration	<code>--by-duration</code> PnDTnHnMnS	Duration must be specified in ISO8601 format. This option will subtract the duration to the current timestamp in the server, and find the offsets using that subtracted timestamp, and reset to those offsets. The duration specified won't consider daylight saving effects.	Reset to first offset since one week ago (from current timestamp): <code>--application-id app1 --input-topics foo,bar --by-duration P7D</code>
3.	Reset to Earliest (DEFAULT)	<code>--to-earliest</code>	This option will reset offsets to the earliest using Kafka Consumer's <code>`#seekToBeginning`</code> . This scenario will be defined by default if none scenario is specified.	Reset to earliest offset available: <code>--application-id app1 --input-topics foo,bar --to-earliest</code>

4.	Reset to Latest	--to-latest	This option will reset offsets to the earliest using Kafka Consumer's <code>`#seekToEnd`</code>	Reset to latest offset available: <code>--application-id app1 --input-topics foo,bar --to-latest</code>
5.	Reset to Offset	--to-offset	This option will reset offsets to an specific value.	Reset to offset 1 in all partitions: <code>--application-id app1 --input-topics foo,bar --to-offset 1</code>
6.	Shift Offset by 'n'	--shift-by n	This option will add the <code>`n`</code> value to the current offset, and reset to the result. <code>`n`</code> can be a positive or negative value, so offset will be move backward if it is negative, and forward if it is positive. If current offset + n is higher than the <i>latest</i> offset, new offset will be set to <i>latest</i> . If current offset + n is lower than the <i>earliest</i> offset, new offset will be set to <i>earliest</i> .	Reset to current offset plus 5 positions: <code>--application-id app1 --input-topics foo,bar --shift-by 5</code>
7.	Shift from file	--from-file PATH_TO_FILE	This option will take a Reset Plan CSV file with the offsets to reset by topics/partitions. It does not require scope, because topics and partitions are defined in the file. A validation will be done to the topics included in the file: it cannot be also included on the <code>`Intermediate`</code> and <code>`Internal`</code> topics managed by the tool.	Reset using a file with reset plan: <code>--application-id app1 --from-file reset-plan.csv</code>

Execution Options

1.	Execute	(no execution arguments)	This execution option will run the reset offset process based on scenario and scope.	Prints and execute resetting all topics and partitions to earliest: <code>--application-id app1 --input-topics foo,bar --to-earliest</code>
2.	Dry-Run	--dry-run	This execution option will only print out the result of the scenario by scope. (i.e. dry-run) The output will look like this: INPUT <div> <div> <div>TOPIC</div> <div>END-OFFSET</div> <div>input-topic</div> <div>100</div> </div> <div> <div>PARTITION</div> <div>0</div> </div> <div> <div>NEW-OFFSET</div> <div>90</div> </div> <div> <div>NEW-LAG</div> <div>10</div> </div> <div>LOG-</div> </div> INTERMEDIATE <div> <div>TOPIC</div> <div>END-OFFSET</div> <div>intermediate-topic</div> <div>100</div> </div> <div> <div>PARTITION</div> <div>0</div> </div> <div> <div>NEW-OFFSET</div> <div>0</div> </div> <div> <div>NEW-LAG</div> <div>100</div> </div> <div>LOG-</div> INTERNAL <div> <div>TOPIC</div> <div>END-OFFSET</div> <div>internal-topic</div> <div>0</div> </div> <div> <div>PARTITION</div> <div>0</div> </div> <div> <div>NEW-OFFSET</div> <div>0</div> </div> <div> <div>NEW-LAG</div> <div>0</div> </div> <div>LOG-</div>	Prints result of resetting all topics and partitions to earliest: <code>--application-id app1 --input-topics foo,bar --to-earliest --dry-run</code>

Proposed Changes

- Implement reset-offset *Scenarios* option described above on ``kafka-streams-application-reset`` using ``KafkaConsumer``.
- Use ``kafka-consumer-groups`` ``reset-offsets`` format to define input topics (**e.g. ``topic2:0,1,2``**) on ``kafka-streams-application-reset``.
- Keep execution parameters uniform between both tools: It will execute by default, and have a ``dry-run`` parameter just show the results. This will involve change current ``ConsumerGroupCommand`` to change execution options.

Compatibility, Deprecation, and Migration Plan

What impact (if any) will there be on existing users?

- Proposed Change 3. breaks compatibility for tools that rely on the ``--execute`` parameter to run the reset operation.
- Other changes should not impact as defaults should make code using the class ``StreamResetter.java`` work as it used to.

If we are changing behavior how will we phase out the older behavior?

- Current option ``--execute`` will be marked as "goint-to-be-deprecated" and replaced by ``dry-run`` parameter, until it gets replaced.

When will we remove the existing behavior?

- Next major release.

Rejected Alternatives

~~Current `kafka-streams-application-reset` as mentioned in the Motivation supports this functionality partially, but it will be deprecated.~~

We will keep both tools `kafka-streams-application-reset` and `reset-offsets`, reusing `reset-offsets` on the background when `kafka-streams-application-reset` is executed.

This KIP was considering to remove Zookeeper dependency, but this has been handled by [KIP-198](#).

[1] [KIP-122: Add Reset Consumer Group Offsets tooling](#)

[2] <https://www.confluent.io/blog/data-reprocessing-with-kafka-streams-resetting-a-streams-application/>