

KIP-1005: Expose EarliestLocalOffset and TieredOffset

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

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: *"Accepted"*

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

Describe the problems you are trying to solve.

[KIP-405: Kafka Tiered Storage](#) introduced the concept of a local log start offset. The local log start offset is the offset of a log above which reads are guaranteed to be served from the disk of the leader broker. In other words, this offset denotes the position after which data is still local. This offset was exposed as part of the ListOffsets API with designated integer -4.

There is a Kafka command line tool called kafka-get-offsets. It is not currently exposing -4 as a valid "reserved" integer with a special meaning i.e. getting the earliest local offset. The purpose of this KIP is to expose this functionality explicitly via the kafka-get-offsets tool.

Internal to the UnifiedLog of a Kafka broker there is one more offset - the highest offset of data stored in remote storage. In other words, this offset denotes the latest offset stored in remote storage. This KIP also aims to expose this offset as designated integer -5 and extend the kafka-get-offsets tool with it as well.

Public Interfaces

Briefly list any new interfaces that will be introduced as part of this proposal or any existing interfaces that will be removed or changed. The purpose of this section is to concisely call out the public contract that will come along with this feature.

The current kafka-get-offsets --time flag has the following description:

```
--time <String: <timestamp> / -1 or latest / -2 or earliest / -3 or max-timestamp
timestamp of the offsets before that.
[Note: No offset is returned, if the timestamp greater than recently committed record timestamp is given.] (default: latest)
```

After the change proposed in this KIP the description will be:

```
--time <String: <timestamp> / -1 or latest / -2 or earliest / -3 or max-timestamp / -4 or earliest-local / -5 or latest-tiered
timestamp of the offsets before that.
[Note: No offset is returned, if the timestamp greater than recently committed record timestamp is given.] (default: latest)
```

In order to achieve this we will be adding new OffsetSpecs [1] - in particular entries for EarliestLocalSpec and LatestTieredSpec.

As part of this KIP we will change how the ListOffsets API behaves. We won't add any fields, but we will introduce a new behaviour on the broker when a timestamp with value of -5 is requested. Going forward, when provided with a timestamp of -5 a broker will return the latest tiered offset. We don't need to add -4 as it has already been added as part of previous KIPs.

The behaviour of asking for the earliest-local timestamp (-4) when Tiered Storage is not enabled is to return the earliest timestamp (-2). We need to define the behaviour of asking for the latest-tiered timestamp (-5) when Tiered Storage is not enabled. **In such a situation, we won't return any offset.** This behaviour is what we do already if someone asks for a timestamp greater than that of a recently committed record (and would thus ask for something outside the bounds of the log).

[1] <https://kafka.apache.org/36/javadoc/org/apache/kafka/clients/admin/OffsetSpec.html>

Proposed Changes

Describe the new thing you want to do in appropriate detail. This may be fairly extensive and have large subsections of its own. Or it may be a few sentences. Use judgement based on the scope of the change.

As per the reference implementation (<https://github.com/apache/kafka/pull/14788>), we would need to add new offset specifications.

Compatibility, Deprecation, and Migration Plan

- *What impact (if any) will there be on existing users?*
- *If we are changing behavior how will we phase out the older behavior?*
- *If we need special migration tools, describe them here.*
- *When will we remove the existing behavior?*

The purpose of this change is to expose broker behaviour already changed as part of another KIP we do not foresee any changes to existing behaviour requiring phasing out or deprecation.

The second purpose is to expose new behaviour, which again, wouldn't require phasing out or deprecating any existing one.

Test Plan

Describe in few sentences how the KIP will be tested. We are mostly interested in system tests (since unit-tests are specific to implementation details). How will we know that the implementation works as expected? How will we know nothing broke?

We will add unit and integration tests where appropriate to demonstrate that the correct offsets are being returned.

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.

We rejected not exposing these offsets. Even though we do not anticipate that customers will frequently ask for them, we believe that making them public will avoid confusion and will aid debugging should problems be encountered while operating Tiered Storage.