

KIP-199: Add Kafka Connect offset tool

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
 - [Exporting](#)
 - [Importing](#)
 - [Override or remove the source offset for one or more source partitions](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

Status

Current state: Discarded - Superseded by [KIP-875](#)

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

Kafka Connect framework stores the most recent offsets for source and sink connectors in Kafka topics. When operating a connector, it is sometimes desirable to manually change or override the persisted offsets. For example, if a connector fails to produce/consume a message at a particular offset, an operator may choose to skip over that source-specific offset and have the connector restart at the next message.

Kafka Connect uses normal Kafka consumers' ability to track their own offsets for *sink connectors*, so it is possible (albeit not ideal) to use the existing Kafka consumer offsets tool to view and manage those offsets. However, Kafka Connect uses a completely different mechanism for source connectors that uses similar "partition" and "offset" terminology except that the partitions and offsets are both generic maps of keys and values defined by the source connector. Kafka Connect stores these offsets either in a local file (for standalone mode) or in Kafka topics (for distributed mode), and there currently is no mechanism provided to read or update these source offsets.

The proposed tool will initially allow users to **read** the existing partition-offset pairs for the connectors, update/modify them, and then **store** the partition-offset pairs back to the Kafka topic. However, over time we expect that this tool will also be improved/expanded to support *sink connector* offsets as well, and this has influenced the design and name of the proposed tool. This tool is also designed to be used **offline** while the workers and connectors are not running; this proposal does not define the behavior of the tool if used while the workers and connectors are running.

Public Interfaces

A new command line tool called `connect-offsets.sh` that will accept several command line parameters. The tool will have two main modes:

1. Exporting the existing partition-offset pairs and writing in JSON format to standard out or to a specified file; and
2. Importing the partition-offset pairs specified in JSON format read from standard in or a specified file

The mode is specified using either the `--export` or `--import` command line parameters, as discussed below. If neither the `--export` or `--import` parameter are specified or if any of the parameters are used incorrectly, the tool will display a usage/help statement that lists the parameters and their descriptions.

NOTE: Any connectors whose offsets are modified by this tool should not be running when this tool is used. This is because while Connect records the offsets periodically, the offsets are **read** only when connectors are started. Although the tool can be run while the connectors are running, doing so means that the connectors will likely overwrite any updated offsets.

Exporting

The tool will read all of the persisted partition-offset pairs and write them out in JSON form to standard out. The usage is as follows:

```
bin/connect-offsets.sh [options]
```

where the options for export mode are as follows:

| Parameter | Required | Description |
|-----------------------|----------|---|
| <code>--config</code> | yes | Specifies the path to the Connect worker configuration. The configuration is used to know how to access the offset storage and the converters used to serialize and deserialize the offset information. |

| | | |
|-------------------------------|----|--|
| <code>--connectors</code> | no | Specifies a comma-separated list of the connector names used to filter which partition-offset pairs are to be exported. |
| <code>--from-beginning</code> | no | Include all of the partition-offset pairs, including those that have been since overwritten by new pairs. If this option is excluded, the tool only outputs the latest partition-offset pairs. |

For example, the following command will read the Connect worker configuration in the `my-worker-config.properties` file and write out all persisted partition-offset pairs to standard out:

```
bin/connect-offsets.sh --config=my-worker-config.properties
```

A sample of this output might be:

```
{
  "MyConnector": [{
    "partition": {
      "file": "a"
    },
    "offset": {
      "offsetKey1": "offsetValue1",
      "offsetKey2": "offsetValue2"
    }
  }, {
    "partition": {
      "file": "b"
    },
    "offset": {
      "offsetKey1": "offsetValue3",
      "offsetKey2": "offsetValue4"
    }
  }]
}
```

Note how the response is a JSON document with field names that correspond to the connector names. The value of each field is an array containing zero or more documents each with a "partition" field and an "offset" field. The "partition" field contains a document with the string-valued fields representing the source-specific partition information, and the "offset" field contains a document with the string-valued fields representing the source-specific offset information.

The `--connectors` parameter takes a comma-separated list of connector names, and will output only the connector offset document that match one of the named connectors:

```
bin/connect-offsets.sh --config=my-worker-config.properties --connectors=MyConnector,OtherConnector
```

The tools JSON output can be piped to a file using standard techniques:

```
bin/connect-offsets.sh --config=my-worker-config.properties --connectors=MyConnector,OtherConnector > my-offsets.json
```

Importing

The tool will read all of the persisted partition-offset pairs supplied on the standard input, and write them to the offset storage. Typically this will involve piping the JSON offset information from a file that was output using the export mode as described above:

```
bin/connect-offsets.sh [options] < my-offsets.json
```

where the options for import mode are as follows:

| Parameter | Required | Description |
|------------------------|----------|---|
| <code>--config</code> | yes | Specifies the path to the Connect worker configuration. The configuration is used to know how to access the offset storage and the converters used to serialize and deserialize the offset information. |
| <code>--dry-run</code> | no | Defaults to "false". When set to "true", the tool will write out the actions that would be performed but will not actually modify any persisted partition-offset pairs. |

Override or remove the source offset for one or more source partitions

A user can use the tool as mentioned above to obtain a copy of the partition-offset pairs. The user can modify the JSON, and send the updated JSON to the tool to update the persisted partition-offset pairs that appear in the file. For example, the following will read the partition-offset pairs in the specified file and update only those partition-offset pairs:

```
bin/connect-offsets.sh --config=my-worker-config.properties < my-offsets.json
```

This tool overwrites the partitions and offsets for the connectors as specified in the input. Any persisted partition-offset pair for a connector not included in the input will be left unmodified.

Note that in addition to modifying the persisted partition-offset pairs, the tool can also be used to *remove* partition offset pairs when the "offset" object in the JSON is null. For example, consider the following input supplied to the tool:

```
{
  "MyConnector": [{
    "partition": {
      "file": "a"
    },
    "offset": {
      "offsetKey1": "offsetValue1",
      "offsetKey2": "offsetValue2"
    }
  }, {
    "partition": {
      "file": "b"
    },
    "offset": null
  }]
}
```

The tool will **update** the "MyConnector" connector's offsets for the partition with file "a" but will **remove** the offsets for the partition with file "b". Any other partition-offset pairs for this or any other connector will be unmodified.

Proposed Changes

Add the tool as described in the Public API section, with initial support only for reading and writing source connector offsets.

Compatibility, Deprecation, and Migration Plan

This proposal adds a new tool and changes no existing functionality.

Rejected Alternatives

One option considered was to have the tool parameters specify the group ID, broker bootstrap servers, topic name, and other connection information. However, this would only have worked for Connect's distributed mode, whereas the proposed approach will work for both Standalone and Distributed mode configurations.