KIP-381: Connect: Tell about records that had their offsets flushed in callback

Status

Current state: *Under Discussion*

Discussion thread: here

JIRA: here

Pull Request: here

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

Motivation

Implementing a Kafka-Connect SourceTask I need exact information about which polled records had their offsets flushed. If I do not get that information, I am not able to keep correct state.

Public Interfaces

SourceTask abstract class (interface) will change from

```
public abstract class SourceTask implements Task {
    ...
    public void commit() throws InterruptedException {
        // This space intentionally left blank.
    }
    ...
    public void commitRecord(SourceRecord record) throws InterruptedException {
        // This space intentionally left blank.
    }
    ...
}
```

to

```
public abstract class SourceTask implements Task {
    ...
    @Deprecated
    public void commit() throws InterruptedException {
        // This space intentionally left blank.
    }
    public void offsetsFlushedAndAcked(List<SourceRecord> offsetsFlushed) throws InterruptedException {
        commit();
    }
    ...
    @Deprecated
    public void commitRecord(SourceRecord record) throws InterruptedException {
        // This space intentionally left blank.
    }
    public void recordSentAndAcked(SourceRecord record) throws InterruptedException {
        commitRecord(record);
    }
    ...
}
```

Besides that JavaDoc descriptions will also change for the better.

Proposed Changes

Implementing a Kafka-Connect SourceTask. There is a callback called "commit", which is called then offsets of polled records have been flushed and acknowledged. Currently this callback does NOT tell for which records the offsets have been flushed. Guess people have used it, assuming that all records that have been polled at the time of callback to "commit", have also had their offsets committed. But that is not true. Implementing a SourceTask I want to know the records that had their offsets flushed. It is obvious to "tell me that" via an argument to the callback.

The name of the callback "commit" is not a very good name for a method telling me that "offsets was committed acknowledged". The name was probably chosen as an indication of what the SourceTask ought to do in the callback. I will argue that there may be lots of different things you want to do in that callback, and that lots of them has nothing to do with commit. What you want to do really depends on the nature of your SourceTask. Therefore I suggest changing the name from "commit" (incorrectly stating what you need to do in the callback) to "offsetsFlushedAndAcked" (stating what you need to react on in the callback). I suggest changing the naming in a backwards-compatible way - deprecating the old method for now.

For consistency, I suggest to also change the naming of the other callback on SourceTask. I suggest changing its name from "commitRecord" to "recordSentAndAcked". Also in a backwards-compatible way.

Compatibility, Deprecation, and Migration Plan

- As far as I can see there is 100% backwards-compatibility. No one needs to do anything to adobt
- A little bit unsure if existing SourceTask implementations (subclasses of SourceTask) needs to be recompiled with the new changed SourceTask in order to work with it - but I guess not
- At same point we may want to remove the "old" deprecated callbacks - or can keep them forever

Rejected Alternatives

It has been suggested (both on JIRA and on mailing-list discussion thread) to just remove the "commit" callback method, with the argument that nobody needs it. I (steff1193) would really like to keep it, because **I** need it in some of the source-connectors I have written. On the Pull Request matthedude also insinuate that he/she would like to keep it - but make it "work" in a predictable manner. Besides that, if we remove the "commit" callback method, the patch will not be backwards compatible.