KIP-440: Extend Connect Converter to support headers

Status

Current state: *Accepted (*Voting thread*)*

Discussion thread: [DISCUSS] KIP-440: Extend Connect Converter to support headers

JIRA: KAFKA-7273

PR: https://github.com/apache/kafka/pull/6362

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

Motivation

Kafka message headers is a simple and straightforward way to enrich Kafka messages with additional metadata. Metadata can be anything - user information, tracing, enriched fields, etc.

When using message schemas, Kafka message headers is the best way to transfer the information about a schema, for example schema ID, because in this case consumer of data doesn't need to deserialize the whole payload to get the schema information using some sort of an envelope. The absence of an envelope or a custom protocol format makes it easy to transfer "raw" payload as is, for example serialized JSON, Avro or Protocol Buffers message. Even if message schemas are not used, something like content type or compression algorithm can be valid header metadata that helps serialization / deserialization.

Kafka Serializer and Deserializer interfaces provide a way to leverage header information in this way. However, Kafka Connect Converter interface does not. It makes it impossible to use Kafka Connect together with Kafka Producers, Consumers or Streams that rely on headers for serialization / deserialization.

This KIP proposes to bring a very similar header support to Kafka Connect.

Public Interfaces

Two new *default* methods will be added to the existing converter interface (connect/api/src/main/java/org/apache/kafka/connect/storage/Converter.java):

```
import org.apache.kafka.common.header.Headers;

default byte[] fromConnectData(String topic, Headers headers, Schema schema, Object value) {
    return fromConnectData(topic, schema, value);
}

default SchemaAndValue toConnectData(String topic, Headers headers, byte[] value) {
    return toConnectData(topic, value);
}
```

Proposed Changes

Converter interface will have two new default methods described in the *Public Interfaces* section (above).

WorkerSinkTask will use the new *toConnectData* method and pass "raw" Kafka message headers. Headers converter is not required in this case.

WorkerSourceTask will use the new *fromConnectData* method and pass the headers received from the headers converter, *RecordHeaders* (which implements *Headers* interface). Headers converter is used as a way to get headers when converting data from internal Connect format to Kafka messages.

Both *WorkerSinkTask* and *WorkerSourceTask* apply these methods for message keys and message values.

Compatibility, Deprecation, and Migration Plan

New interface methods are default methods, which means all existing implementations won’t need any changes.

Rejected Alternatives

Headers converter interface has been introduced in KIP-145 and it provides a mechanism to serialize and deserialize header values.

Unfortunately, using headers converter alone to solve the problem defined in the *Motivation* section (above) is not enough, headers converter is only used for headers transformation. It doesn't provide a way to apply this kind of transformation to message keys and values.