Support for custom Kafka headers (PJ48074 – February 2025)

JAM Services

Dan Gritter

2025 TPF Users Group Conference
May 5-7, Austin, TX

IBM Z

IBM

Problem Statement

Guaranteed delivery requests, sent with the tpf_publish_data API, do not have the ability to send custom Kafka headers. There is currently no way to send additional metadata along with requests, without including it as part of the message data.

As-Is User Story

Headers

A developer runs a z/TPF JAM that sends messages to an Apache Kafka consumer. A new directive requires correlation identifiers to be included with each message. The identifiers are included in the data of the message and sent to Kafka. When the messages are consumed, the identifiers must be separated from the message data, increasing the complexity of message processing.

Value Statement

Headers

The developer can use custom Kafka headers to send metadata like correlation identifiers with guaranteed delivery messages. Headers are stored separately from the message data, allowing for easy access on the consumer side. This can reduce the complexity of data parsing on the Kafka consumer end, saving processing time.

To-Be User Story

Headers

The developer runs a z/TPF JAM that sends messages to an Apache Kafka consumer. Custom correlation identifiers are created to attach to guaranteed delivery messages. Using custom Kafka headers to store these identifiers enables the Kafka consumer to easily access and parse these identifiers for each message.

As-Is User Story

User data

A developer runs a z/TPF JAM that sends messages to an Apache Kafka consumer. Tracking errors when messages are put on the error queue by the z/TPF JAM can be difficult. Correlating the error to the specific message takes time and expertise.

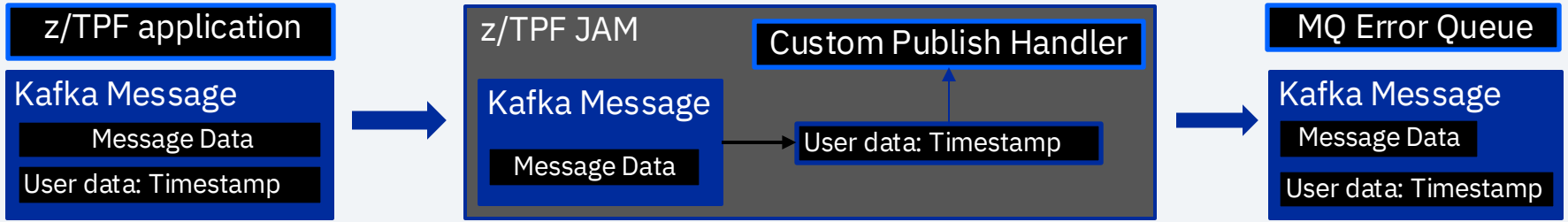
Value Statement

User data

The developer can leverage the custom user data field that can be sent with guaranteed delivery messages to provide contextual information, like timestamps. The user data field is intended to be used on z/TPF to give additional information about each individual message when processing messages on the error queue.

To-Be User Story

User data



The developer runs a z/TPF JAM that sends messages to an Apache Kafka consumer. A timestamp for creation of the message or time of the publish API call is added using the custom user data field. This allows for the message to have the timestamp associated with the message in the publish handler. When errors occur, the timestamp tracks when the problem message was created or sent.

Technical Details

tpf_publish_data_ext API

pub_args – custom header linked list

user_data – custom metadata field, e.g. timestamp

Both new fields are optional, and the old tpf_publish_data API is still supported.

```
int tpf_publish_data_ext(
                char * targetName,
                char * topic_data,
                char * key_data,
                unsigned short key_len,
                char * data,
                unsigned int data_len,
                publish_data_args *pub_args,
                char * user_data
)
```

Technical Details

publish_data_args structure

- Header must be passed to the publish API using this structure.
- Linked list format
- Key-value pairs and their lengths.
- **nextheader** pointer to next header, if last should be NULL

```
struct publish_data_args {
        publish_data_args *nextheader;
        unsigned short headerKeyLen;
        unsigned short headerValueLen;
        char       *headerKey;
        char       *headerValue;
}
```

Technical Details

Viewing Queue Contents

Kafka headers and user data are shown in the ZJAMC display *queue* display. If data is not in UTF8 display will show in Hex.

```
ZJAMC DISPLAY INP N-TPFKAFKA
START OF ZJAMC QUEUE DISPLAY FOR INPUT.QUEUE
45 MESSAGES ON QUEUE
INPUT.QUEUE(0):
topicName=flightStatus
key=flight34
data=seats93departure1101airtime1233
headers=flightnumber34airline23
userdata=timestamp 1234145212425590
```

```
ZJAMC DISPLAY INP N-TPFKAFKA
START OF ZJAMC QUEUE DISPLAY FOR INPUT.QUEUE
45 MESSAGES ON QUEUE
INPUT.QUEUE(0):
THE ZJAMC DISPLAY COMMAND COULD NOT FORMAT A QUEUE
ENTRY.
Translation error occurred converting UTF8 to IBM1047
- SWITCHING TO HEX DATA _
topicName=666C69676874537461747573
key=666C696768743334
data=736561747339336465706172747572653131303161697 2..
headers=666C696768746E756D6265723346169726C696E653..
userdata=74696D657374616D702031323334313435323132 34..
```

Conclusion

Kafka headers are key-value pairs that you can attach to Apache Kafka messages to provide additional metadata or information about the message.

Headers are stored separately from the message data, allowing for easy access on the consumer side. By storing additional metadata on Kafka messages, you can correlate or filter the messages based on the metadata when the Kafka consumer receives them.

User data can be used to enhance processing of messages on z/TPF in the event an unrecoverable error occurs.

Thank you