

# KIP-132: Augment KStream.print to allow extra parameters in the printed string

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)

## Status

**Current state:** *Discarded (duplicated by [KIP-160](#))*

**Discussion thread:**

**JIRA:** [here](#)

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

## Motivation

As for now, `KStream#print` leads to a predefined output where `key` and `value` are printed with comma separation:

```
"[" + this.streamName + "]: " + keyToPrint + " , " + valueToPrint
```

We therefore suggest to extend `print` in a way that it takes `KeyValueMapper` as a parameter, to let the user modify the output according to its needs. If a mapper is present, the hard-coded key-value output can be replaced with:

```
"[" + this.streamName + "]: " + mapper.apply(keyToPrint, valueToPrint)
```

## Public Interfaces

The affected interface is `KStream`, which needs to be extended with another overloaded version of `print`:

```
void print(final Serde<K> keySerde,
           final Serde<V> valSerde,
           final String streamName,
           final KeyValueMapper<K, V, String> mapper);
```

## Proposed Changes

Apart from the extension of the interface, the previous implementation of `print` will be call a new overloaded version thereof. In case no `KeyValueMapper` is set, `null` serves as the indicator to use the default output:

```
@Override
public void print(Serde<K> keySerde, Serde<V> valSerde, String streamName) {
    print(keySerde, valSerde, streamName, null);
}

@Override
public void print(final Serde<K> keySerde, final Serde<V> valSerde, String streamName, final KeyValueMapper<K,
V, String> mapper){
    ...
}
```

# Compatibility, Deprecation, and Migration Plan

The extension of `print` will not introduce compatibility issues. Further more, we can maintain the current output by keeping the current output format as a default (if mapper was not set):

```
if(mapper == null) {
    printStream.println "[" + streamName + "]: " + keyToPrint + " , " + valueToPrint);
} else {
    printStream.println "[" + streamName + "]: " + mapper.apply(keyToPrint, valueToPrint));
}
```