

```
}
}
```

任何传递给 HDFS Sink 带有 `serializer` 后缀的配置参数, 会被传递给序列化器。例如, 需要字符集的序列化器可以如下配置:

```
agent.sinks.hdfsSink.serializer.bufferSize = 4096
agent.sinks.hdfsSink.serializer.charset = UTF-8
```

这些参数通过 `Context` 实例进行传递 (只有参数和值被传递——前面的任何东西是不会传递的), 且用来配置序列化器。简单来讲, 序列化器假设事件主体已经编码为合适的格式; 它不会改变且按照该格式写出。

Flume 内置了三类序列化器: `TEXT` 序列化器、`HEADER_AND_TEXT` 序列化器和 `AVRO_EVENT` 序列化器。`TEXT` 序列化器以事件主体本身写入到文件, 可以根据需要在事件之间插入新的一行。`HEADER_AND_TEXT` 序列化器也做相同的事情, 但也包括 “key=value” 格式的事件报头。默认会在事件之间插入新的一行。这可以通过将 `appendNewline` 的值设置为 `false` 来禁用, 如下面的展示:

```
agent.sinks.hdfsSink.fileType = DataStream
agent.sinks.hdfsSink.serializer = HEADER_AND_TEXT
agent.sinks.hdfsSink.serializer.appendNewline = false
```

`AVRO_EVENT` 序列化器使用下面所示的模式, 将事件写出为 Avro container 文件 [container-files]。

```
{
  "type": "record",
  "name": "Event",
  "fields": [
    {
      "name": "headers",
      "type": {
        "type": "map",
        "values": "string"
      }
    },
    {
      "name": "body",
      "type": "bytes"
    }
  ]
}
```