

KIP-659: Improve TimeWindowedDeserializer and TimeWindowedSerde to handle window size

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

Status

Current state: Accepted

Discussion thread: [here](#)

Vote thread: [here](#)

JIRA:

 Unable to render Jira issues macro, execution error.

 Unable to render Jira issues macro, execution error.

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

Motivation

Window size is not passed to the Kafka consumer during windowed aggregations, so the consumer uses the default Long.MAX_VALUE when initializing the deserializer. Both *TimeWindowedSerde* and *TimeWindowedDeserializer* have a default constructor that does not require a window size, causing fatal errors at runtime for some programs.

Public Interfaces

In *StreamsConfig.java* add *window.size.ms*

StreamsConfig

```
/** <code>window.size.ms</code> */
public static final String WINDOW_SIZE_MS_CONFIG = "window.size.ms";
private static final String WINDOW_SIZE_MS_DOC = "Passes window size to the deserializer in order to calculate window end times.";
```

Deprecate the following methods

TimeWindowedSerde

```
@Deprecated
public TimeWindowedSerde(final Serde<T> inner) {
    super(new TimeWindowedSerializer<>(inner.serializer()), new TimeWindowedDeserializer<>(inner.deserializer()));
}
```

WindowedSerdes

```
/**
 * Construct a {@code TimeWindowedSerde} object for the specified inner class type.
 */
@Deprecated
static public <T> Serde<Windowed<T>> timeWindowedSerdeFrom(final Class<T> type) {
    return new TimeWindowedSerde<>(Serdes.serdeFrom(type));
}
```

TimeWindowedDeserializer

```
@Deprecated
public TimeWindowedDeserializer(final Deserializer<T> inner) {
    this(inner, Long.MAX_VALUE);
}
```

Log an error message in *Serdes.scala* for *implicit def timeWindowedSerde* and pass through to the deserializer

Serdes

```
// Log error message and pass through to underlying deserializer
implicit def timeWindowedSerde[T](implicit tSerde: Serde[T]): WindowedSerdes.TimeWindowedSerde[T] =
    new WindowedSerdes.TimeWindowedSerde[T](tSerde)
```

Proposed Changes

Add *window.size.ms* to the *StreamsConfigs* class to ensure that the desired window size is set when the consumer creates the *TimeWindowedDeserializer* instance. This is relevant for consuming *TimeWindowed* records directly, which is helpful when testing windowed aggregations. Without this change, all windows will have a size and end time of *Long.MAX_VALUE*. Users can *either* set the window size through the config *or* set the window size through the constructor. To help enforce this, the *window.size.ms* will not have a default value.

Additionally, deprecate constructors in both *TimeWindowedDeserializer* and *WindowedSerdes* that don't require a window size. This ensures that the window size would be properly set before encountering any instances where it should be used. When these are deprecated, the *timeWindowedSerde* implicit in *Serdes.scala* will be updated to log an error message explaining that this implicit should no longer be used. To keep backwards compatibility, this implicit won't be deprecated but will pass through to the deserializer.

Note that to use the *windows.size.ms* config through the console consumer (if desired), add the *key.deserializer* prefix and pass it in as a property.

Compatibility, Deprecation, and Migration Plan

Tests that need to be updated:

- [WindowKeySchemaTest](#)
- [KStreamAggregationIntegrationTest](#)
- [WindowedSerdesTest](#)
- [TimeWindowedCogroupedKStreamImplTest](#)

Classes that need to be updated:

- [TemperatureDemo](#)
- [Serdes](#)

Rejected Alternatives