

KIP-319: Replace segments with segmentInterval in WindowBytesStoreSupplier


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

Status

Current state: *Accepted*

Discussion thread: [here](#)

JIRA:

A screenshot of a Jira error message. It features a yellow warning triangle icon on the left, followed by the text "Unable to render Jira issues macro, execution error." in a standard sans-serif font.

PR: <https://github.com/apache/kafka/pull/5257>

Release: 2.1

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

Motivation

The `WindowBytesStore` interface provides a method to get the number of segments in the store so that the `WindowStoreBuilder` can use the information to wrap the store in a cache. The problem is that the cache needs to know the segment **interval**, not the segment **count**.

This KIP proposes to replace the current method with one providing the correct information.

Further, at this point, it seems that most applications will either want to accept the default segment interval (which is scaled to the retention period), or to select a specific size. In support of this, I propose to replace the "segments" nomenclature in Windows with "segmentInterval" as well.

Proposed Public Interface Change

In `WindowBytesStoreSupplier`, we will:

- `deprecate int segments()`
- `add long segmentInterval()`

```

/**
 * The number of segments the store has. If your store is segmented then this should be the number of segments
 * in the underlying store.
 * It is also used to reduce the amount of data that is scanned when caching is enabled.
 *
 * @return number of segments
 * @deprecated since 2.1. Use {@link WindowBytesStoreSupplier#segmentInterval()} instead.
 */
@Deprecated
int segments();

/**
 * The size of the segments (in milliseconds) the store has.
 * If your store is segmented then this should be the size of segments in the underlying store.
 * It is also used to reduce the amount of data that is scanned when caching is enabled.
 *
 * @return size of the segments (in milliseconds)
 */
long segmentInterval();

```

In Windows, we will:

- deprecate segments field (it was unintentionally made public before)
- add a public segmentInterval() method
- deprecate segments(int)

```

+   @Deprecated public int segments = 3;

+   /**
+    * Return the segment interval in milliseconds.
+    *
+    * @return the segment interval
+    */
+   public long segmentInterval();

+   /**
+    * Set the number of segments to be used for rolling the window store.
+    * This function is not exposed to users but can be called by developers that extend this class.
+    *
+    * Note: previously, this would bound the total number of segments in the store, but as of 2.1, it is used
+    solely to determine the segment size. The actual number of segments is a function of how many future events are
+    in flight.
+    *
+    * @param segments the number of segments to be used
+    * @return itself
+    * @throws IllegalArgumentException if specified segments is small than 2
+    * @deprecated since 2.1 Override segmentInterval() instead.
+    */
+   @Deprecated
+   protected Windows<W> segments(final int segments) throws IllegalArgumentException;

```

In Stores, we will:

- deprecate persistentWindowStore() that takes numSegments
- add persistentWindowStore() that takes segmentInterval
- add persistentWindowStore() that doesn't parameterize segments

```

/**
 * Create a persistent {@link WindowBytesStoreSupplier}.
 * @param name name of the store (cannot be {@code null})
 * @param retentionPeriod length of time to retain data in the store (cannot be negative)
 * @param numSegments number of db segments (cannot be zero or negative). Note: previously, this
would bound the total number of segments in the store, but as of 2.1, it is used solely to determine the
segment size. The actual number of segments is a function of how many future events are in flight.
 * @param windowSize size of the windows (cannot be negative)
 * @param retainDuplicates whether or not to retain duplicates.
 * @return an instance of {@link WindowBytesStoreSupplier}
+ * @deprecated since 2.1 Use {@link Stores#persistentWindowStore(String, long, long, boolean, long)}
instead
 */
+ @Deprecated
public static WindowBytesStoreSupplier persistentWindowStore(final String name,
                                                             final long retentionPeriod,
                                                             final int numSegments,
                                                             final long windowSize,
                                                             final boolean retainDuplicates);

+ /**
+ * Create a persistent {@link WindowBytesStoreSupplier}.
+ * @param name name of the store (cannot be {@code null})
+ * @param retentionPeriod length of time to retain data in the store (cannot be negative)
+ * @param windowSize size of the windows (cannot be negative)
+ * @param retainDuplicates whether or not to retain duplicates.
+ * @return an instance of {@link WindowBytesStoreSupplier}
+ */
+ public static WindowBytesStoreSupplier persistentWindowStore(final String name,
+                                                             final long retentionPeriod,
+                                                             final long windowSize,
+                                                             final boolean retainDuplicates);

+ /**
+ * Create a persistent {@link WindowBytesStoreSupplier}.
+ * @param name name of the store (cannot be {@code null})
+ * @param retentionPeriod length of time to retain data in the store (cannot be negative)
+ * @param windowSize size of the windows (cannot be negative)
+ * @param retainDuplicates whether or not to retain duplicates.
+ * @param segmentInterval size of segments in ms (must be at least one minute)
+ * @return an instance of {@link WindowBytesStoreSupplier}
+ */
+ public static WindowBytesStoreSupplier persistentWindowStore(final String name,
+                                                             final long retentionPeriod,
+                                                             final long windowSize,
+                                                             final boolean retainDuplicates,
+                                                             final long segmentInterval);

```

Compatibility, Deprecation, and Migration Plan

The change in this KIP is backward compatible as it only deprecates the existing method.

Rejected Alternatives

None.