KIP-233: Simplify StreamsBuilder#addGlobalStore

Status

Current state: *Accepted*

Discussion thread: *here*

JIRA: KAFKA-6138 *here*

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

Motivation

StreamsBuilder#addGlobalStore does not provide a good user experience as users are forced to specify names for processor names – processor name are a Processor API detail should be hidden in the DSL. The current API is the following:

```
public synchronized StreamsBuilder addGlobalStore(final StoreBuilder storeBuilder,
                                                  final String topic,
                                                  final String sourceName,
                                                  final Consumed consumed,
                                                  final String processorName,
                                                  final ProcessorSupplier stateUpdateSupplier)
```

Public Interfaces

Add a new function in the org.apache.kafka.streams.StreamsBuilder, as the following:

```
public synchronized StreamsBuilder addGlobalStore(final StoreBuilder storeBuilder,
                                                  final String topic,
                                                  final Consumed consumed,
                                                  final ProcessorSupplier stateUpdateSupplier)
```

Proposed Changes

We should remove the two parameters sourceName and processorName in both InternalStreamsBuilder#addGlobalStateStore and StreamsBuilder#addGlobalStateStore. The two parameter will be generated by the following code.

```
final String sourceName = newProcessorName(KStreamImpl.SOURCE_NAME);
final String processorName = newProcessorName(KTableImpl.SOURCE_NAME);
```

This is the same implementation used in InternalStreamsBuilder#globalTable.

The following code is an example of the new API usage.

```
```

```
String globalTopicName = "testGlobalTopic";
String globalStoreName = "testAddGlobalStore";
final StreamsBuilder builder = new StreamsBuilder();
KeyValueStoreBuilder t = new KeyValueStoreBuilder();
final KeyValueStoreBuilder globalStoreBuilder = EasyMock.createNiceMock(KeyValueStoreBuilder.class)
;
EasyMock.expect(globalStoreBuilder.name()).andReturn(globalStoreName).anyTimes();
EasyMock.replay(globalStoreBuilder);
builder.addGlobalStore(globalStoreBuilder,globalTopicName,new ConsumedInternal(),new MockProcessor
Supplier());
```

Compatibility, Deprecation, and Migration Plan

- *To be backward compatible, the current method must be deprecated and a new method should be added with reduced number of parameters.*
- *Upgraded application should stay with old API since the new API will affected naming of generated internal topic. However, new application can use the new API.*