

Breaking changes to (stateful) topology

- Requirements change (or there was a bug)
- Perhaps analogous to a traditional database migration
- Application reset tool
 - Reset/shift/skip input topic offset
 - Delete internal topics
- No way to transform (eg, migrate) internal topic data apart from replaying messages
- Requires careful thought on how to architect a streams application:
 - Replaying messages
 - Side effects
 - Impact on downstream applications

Breaking changes (gotchas)

- Can be difficult to know when you are introducing a breaking change to a streams topology
 - No “versioning” of topologies registered against application id
- Gotcha: if the internal topic exists, and you make a breaking change, Kafka streams will still happily use your existing state store even if data is technically “incorrect”
- This is also true for local RocksDB cache (don’t forget to call the **cleanUp** method!)