

# Knowledge Base for Client and Binding Implementors

## Knowledge Base

Implementing ZooKeeper recipes is non-trivial. There are many undocumented edge cases that must be correctly handled. This wiki will server as a knowledge-base of known edge cases and methods of working-around and/or handling them.

## Table of Contents

- [Connection failures can make creating ephemeral-sequential nodes difficult](#)
- [Connection failures can make deleting nodes difficult](#)
- [Implement a retry mechanism](#)

## Connection failures can make creating ephemeral-sequential nodes difficult

When creating an ephemeral-sequential, clients need to get the actual name of the node created (i.e. the requested ZNode name plus its sequence number). However, if there is an edge case whereby the server will successfully create the ephemeral-sequential node but the client will not get the result (due to partitioning, etc.). If the client reconnects before the session expires, the newly created ephemeral-sequential will still exist and wreak havoc with the recipe.

A good workaround for this issue is to include a GUID/UUID in the ZNode name. If there is a KeeperException during node creation, wait until successful re-connection, call getChildren and search for a ZNode the contains the GUID/UUID in its name. If found, you can be sure that this was the node you previously created.

## Connection failures can make deleting nodes difficult

Most of the ZooKeeper recipes involve creating a ephemeral-sequential node and then deleting that node to signal that another client can take over (etc.). If, while trying to delete the node, there is a network partition, etc. the node deletion will fail. If the client reconnects before the session expires, however, the ephemeral node will not expire.

A good workaround for this is to have a deletion queue/thread. Nodes that need deleting are added to the queue. If a KeeperException is thrown during delete, the node is added back to the queue for eventual deletion.

## Implement a retry mechanism

As the ZooKeeper docs make clear, there are a number of "recoverable" exceptions that clients must deal with. In particular, ConnectionLossException and SessionExpiredException. Good ZooKeeper client applications are designed to account for failure. Your ZooKeeper ensemble **will experience connection problems in production**. Therefore, so your recipes should account for this. A retry mechanism that catches ConnectionLossException, etc. and retries operations as appropriate is highly advised.