

# Index

## General Information

ZooKeeper: Because coordinating distributed systems is a Zoo

- [ZooKeeper Home Page](#)
- [Documentation](#)
- also see: [The Tao of ZooKeeper](#)

ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. All of these kinds of services are used in some form or another by distributed applications. Each time they are implemented there is a lot of work that goes into fixing the bugs and race conditions that are inevitable. Because of the difficulty of implementing these kinds of services, applications initially usually skimp on them, which make them brittle in the presence of change and difficult to manage. Even when done correctly, different implementations of these services lead to management complexity when the applications are deployed.

ZooKeeper aims at distilling the essence of these different services into a very simple interface to a centralized coordination service. The service itself is distributed and highly reliable. Consensus, group management, and presence protocols will be implemented by the service so that the applications do not need to implement them on their own. Application specific uses of these will consist of a mixture of specific components of ZooKeeper and application specific conventions. [ZooKeeper Recipes](#) shows how this simple service can be used to build much more powerful abstractions.

We have Java and C interfaces to ZooKeeper for the applications themselves. A variety of client bindings is available for a number of languages including Python, Ruby and Go.

- [Overview](#) of ZooKeeper
- [Tutorial](#): A crash course on how to implement primitives with ZooKeeper
- [FAQ](#)
- [Client bindings](#)
- [Useful Tools](#)
- [Presentations](#) and [articles](#) about ZooKeeper
- [PoweredBy](#), a list of sites and applications powered by ZooKeeper
- [ErrorHandling](#), how applications and libraries should deal with errors from ZooKeeper
- [Troubleshooting](#), specific help on troubleshooting ZooKeeper operating environments
- [Performance](#), a benchmark of the performance of the ZooKeeper 3.2 release
- [Zab](#), random notes about the ZooKeeper Atomic Broadcast protocol
- [Failure Scenarios](#), How failure scenarios are handled in ZooKeeper ensembles
- [Upgrade FAQ](#), Known issues and workarounds on ZooKeeper upgrading

## Developer Documentation

- [Roadmap](#)
- [HowToContribute](#)
- [HowToRelease](#)
- [ProjectSuggestions](#)
- [ReleaseManagement](#)
- [CommitterCriteria](#)
- [CommitterOnboarding](#)
- [Merging Github Pull Requests](#)
- [UsingEclipse](#)
- [Committing changes](#)
- [June2011DeveloperMeeting](#)
- [ImplementationCriticisms](#)
- [WebSiteSetup](#)

## Related Projects

- [BookKeeper](#)
- [HedWig](#)
- [HBase's integration of ZooKeeper](#)
- [GSoC2010: Failure Detector Model](#)
- [GSoC2010: Monitoring Recipes and Web-based Administrative Interface](#)
- [ZooKeeper Cluster Membership](#)