ZooKeeper 3.0.0 Release Notes

by

Table of contents

These release notes include new developer and user facing incompatibilities, features, and major improvements.

1. Migration Instructions when Upgrading to 3.0.0

You should only have to read this section if you are upgrading from a previous version of ZooKeeper to version 3.0.0, otw skip down to changes

A small number of changes in this release have resulted in non-backward compatible Zookeeper client user code and server instance data. The following instructions provide details on how to migrate code and date from version 2.2.1 to version 3.0.0.

Note: ZooKeeper increments the major version number (major.minor.fix) when backward incompatible changes are made to the source base. As part of the migration from SourceForge we changed the package structure (com.yahoo.zookeeper.* to org.apache.zookeeper.*) and felt it was a good time to incorporate some changes that we had been withholding. As a result the following will be required when migrating from 2.2.1 to 3.0.0 version of ZooKeeper.

1.1. Migrating Client Code

The underlying client-server protocol has changed in version 3.0.0 of ZooKeeper. As a result clients must be upgraded along with serving clusters to ensure proper operation of the system (old pre-3.0.0 clients are not guaranteed to operate against upgraded 3.0.0 servers and vice-versa).

1.1.1. Watch Management

In previous releases of ZooKeeper any watches registered by clients were lost if the client lost a connection to a ZooKeeper server. This meant that developers had to track watches they were interested in and reregister them if a session disconnect event was recieved. In this release the client library tracks watches that a client has registered and reregisters the watches when a connection is made to a new server. Applications that still manually reregister interest should continue working properly as long as they are able to handle unsolicited watches. For

example, an old application may register a watch for /foo and /goo, lose the connection, and reregister only /goo. As long as the application is able to recieve a notification for /foo, (probably ignoring it) the applications does not to be changes. One caveat to the watch management: it is possible to miss an event for the creation and deletion of a znode if watching for creation and both the create and delete happens while the client is disconnected from ZooKeeper.

This release also allows clients to specify call specific watch functions. This gives the developer the ability to modularize logic in different watch functions rather than cramming everything in the watch function attached to the ZooKeeper handle. Call specific watch functions receive all session events for as long as they are active, but will only receive the watch callbacks for which they are registered.

1.1.2. Java API

1. The java package structure has changed from **com.yahoo.zookeeper*** to **org.apache.zookeeper***. This will probably effect all of your java code which makes use of ZooKeeper APIs (typically import statements)
2. A number of constants used in the client ZooKeeper API were re-specified using enums (rather than ints). See ZOOKEEPER-7, ZOOKEEPER-132 and ZOOKEEPER-139 for full details
3. ZOOKEEPER-18 removed KeeperStateChanged, use KeeperStateDisconnected instead

Also see the current java API

1.1.3. C API

1. A number of constants used in the client ZooKeeper API were renamed in order to reduce namespace collision, see ZOOKEEPER-6 for full details

1.2. Migrating Server Data

The following issues resulted in changes to the on-disk data format (the snapshot and transaction log files contained within the ZK data directory) and require a migration utility to be run.

- ZOOKEEPER-27 Unique DB identifiers for servers and clients
- ZOOKEEPER-32 CRCs for ZooKeeper data
- ZOOKEEPER-33 Better ACL management
- ZOOKEEPER-38 headers (version+) in log/snap files

The following must be run once, and only once, when upgrading the ZooKeeper server instances to version 3.0.0.

> **Note:**
>
> The <dataLogDir> and <dataDir> directories referenced below are specified by the *dataLogDir* and *dataDir* specification in your ZooKeeper config file respectively. *dataLogDir* defaults to the value of *dataDir* if not specified explicitly in the ZooKeeper server config file (in which case provide the same directory for both parameters to the upgrade utility).

1. Shutdown the ZooKeeper server cluster.
2. Backup your <dataLogDir> and <dataDir> directories
3. Run upgrade using
   - `bin/zkServer.sh upgrade <dataLogDir> <dataDir>`

   or
   - `java -classpath pathtolog4j:pathtozookeeper.jar UpgradeMain <dataLogDir> <dataDir>`

   where <dataLogDir> is the directory where all transaction logs (log.*) are stored. <dataDir> is the directory where all the snapshots (snapshot.*) are stored.
4. Restart the cluster.

If you have any failure during the upgrade procedure keep reading to sanitize your database.

This is how upgrade works in ZooKeeper. This will help you troubleshoot in case you have problems while upgrading

1. Upgrade moves files from <dataLogDir> and <dataDir> to <dataLogDir>/version-1/ and <dataDir>/version-1 respectively (version-1 sub-directory is created by the upgrade utility).
2. Upgrade creates a new version sub-directory <dataDir>/version-2 and <dataLogDir>/version-2
3. Upgrade reads the old database from <dataDir>/version-1 and <dataLogDir>/version-1 into the memory and creates a new upgraded snapshot.
4. Upgrade writes the new database in <dataDir>/version-2.

Troubleshooting.

1. In case you start ZooKeeper 3.0 without upgrading from 2.0 on a 2.0 database - the servers will start up with an empty database. This is because the servers assume that <dataDir>/version-2 and <dataLogDir>/version-2 will have the database to start with.

Since this will be empty in case of no upgrade, the servers will start with an empty database. In such a case, shutdown the ZooKeeper servers, remove the version-2 directory (remember this will lead to loss of updates after you started 3.0.) and then start the upgrade procedure.

2. If the upgrade fails while trying to rename files into the version-1 directory, you should try and move all the files under <dataDir>/version-1 and <dataLogDir>/version-1 to <dataDir> and <dataLogDir> respectively. Then try upgrade again.
3. If you do not wish to run with ZooKeeper 3.0 and prefer to run with ZooKeeper 2.0 and have already upgraded - you can run ZooKeeper 2 with the <dataDir> and <dataLogDir> directories changed to <dataDir>/version-1 and <dataLogDir>/version-1. Remember that you will lose all the updates that you made after the upgrade.

1.3. Migrating Server Configuration

There is a significant change to the ZooKeeper server configuration file.

The default election algorithm, specified by the *electionAlg* configuration attribute, has changed from a default of *0* to a default of *3*. See Cluster Options section of the administrators guide, specifically the *electionAlg* and *server.X* properties.

You will either need to explicitly set *electionAlg* to it's previous default value of *0* or change your *server.X* options to include the leader election port.

2. Changes Since ZooKeeper 2.2.1

Version 2.2.1 code, documentation, binaries, etc... are still accessible on SourceForge

Issue	Notes
ZOOKEEPER-1	notes, placeholder - TBD during release process

Table 1: Changes Since ZooKeeper 2.2.1