

# ZooKeeper JMX

by

## Table of contents

|  |   |
|--|---|
| 1 JMX.....                                 | 2 |
| 2 Starting ZooKeeper with JMX enabled..... | 2 |
| 3 Run a JMX console.....                   | 2 |
| 4 ZooKeeper MBean Reference.....           | 3 |

## 1 JMX

Apache ZooKeeper has extensive support for JMX, allowing you to view and manage a ZooKeeper serving ensemble.

This document assumes that you have basic knowledge of JMX. See [Sun JMX Technology](#) page to get started with JMX.

See the [JMX Management Guide](#) for details on setting up local and remote management of VM instances. By default the included *zkServer.sh* supports only local management - review the linked document to enable support for remote management (beyond the scope of this document).

## 2 Starting ZooKeeper with JMX enabled

The class *org.apache.zookeeper.server.quorum.QuorumPeerMain* will start a JMX manageable ZooKeeper server. This class registers the proper MBeans during initialization to support JMX monitoring and management of the instance. See *bin/zkServer.sh* for one example of starting ZooKeeper using *QuorumPeerMain*.

## 3 Run a JMX console

There are a number of JMX consoles available which can connect to the running server. For this example we will use Sun's *jconsole*.

The Java JDK ships with a simple JMX console named [jconsole](#) which can be used to connect to ZooKeeper and inspect a running server. Once you've started ZooKeeper using *QuorumPeerMain* start *jconsole*, which typically resides in *JDK\_HOME/bin/jconsole*

When the "new connection" window is displayed either connect to local process (if *jconsole* started on same host as Server) or use the remote process connection.

By default the "overview" tab for the VM is displayed (this is a great way to get insight into the VM btw). Select the "MBeans" tab.

You should now see *org.apache.ZooKeeperService* on the left hand side. Expand this item and depending on how you've started the server you will be able to monitor and manage various service related features.

Also note that ZooKeeper will register log4j MBeans as well. In the same section along the left hand side you will see "log4j". Expand that to manage log4j through JMX. Of particular interest is the ability to dynamically change the logging levels used by editing the appender and root thresholds. Log4j MBean registration can be disabled by passing - *Dzookeeper.jmx.log4j.disable=true* to the JVM when starting ZooKeeper.

## 4 ZooKeeper MBean Reference

This table details JMX for a server participating in a replicated ZooKeeper ensemble (ie not standalone). This is the typical case for a production environment.

| MBean                | MBean Object Name      | Description   |
|----------------------|------------------------|---|
| Quorum               | ReplicatedServer_id<#> | Represents the Quorum, or Ensemble - parent of all cluster members. Note that the object name includes the "myid" of the server (name suffix) that your JMX agent has connected to.   |
| LocalPeer RemotePeer | replica.<#>            | Represents a local or remote peer (ie server participating in the ensemble). Note that the object name includes the "myid" of the server (name suffix).   |
| LeaderElection       | LeaderElection         | Represents a ZooKeeper cluster leader election which is in progress. Provides information about the election, such as when it started.  |
| Leader               | Leader                 | Indicates that the parent replica is the leader and provides attributes/operations for that server. Note that Leader is a subclass of ZooKeeperServer, so it provides all of the information normally associated with a ZooKeeperServer node.   |
| Follower             | Follower               | Indicates that the parent replica is a follower and provides attributes/operations for that server. Note that Follower is a subclass of ZooKeeperServer, so it provides all of the information normally associated with a ZooKeeperServer node. |
| DataTree             | InMemoryDataTree       | Statistics on the in memory znode database, also operations to access finer (and more computationally intensive) statistics on the data (such as ephemeral count).  |

| MBean      | MBean Object Name | Description   |
|------------|-------------------|---|
|            |                   | InMemoryDataTrees are children of ZooKeeperServer nodes.  |
| ServerCnxn | <session_id>      | Statistics on each client connection, also operations on those connections (such as termination). Note the object name is the session id of the connection in hex form. |

Table 1: MBeans, their names and description

This table details JMX for a standalone server. Typically standalone is only used in development situations.

| MBean           | MBean Object Name        | Description   |
|-----------------|--------------------------|---|
| ZooKeeperServer | StandaloneServer_port<#> | Statistics on the running server, also operations to reset these attributes. Note that the object name includes the client port of the server (name suffix).            |
| DataTree        | InMemoryDataTree         | Statistics on the in memory znode database, also operations to access finer (and more computationally intensive) statistics on the data (such as ephemeral count).      |
| ServerCnxn      | <session_id>             | Statistics on each client connection, also operations on those connections (such as termination). Note the object name is the session id of the connection in hex form. |

Table 2: MBeans, their names and description